A Novel Comprehensive Quality Evaluation Model in Automated Semantic Web Service Composition

Chen Wang, Hui Ma, and Aaron Chen

School of Engineering and Computer Science,
Victoria University of Wellington, New Zealand
{Chen.Wang, Hui.Ma, and Aaron.Chen}@ecs.vuw.ac.nz

Abstract. Semantic web service composition has been a prevailing research area in recent years. There are two major governance challenges faced by researchers, and Semantic matchmaking is to discover interoperable web services, which demands a rich machine understanding language such as OWL-S (Web Ontology language for Web Services), WSML (Web Service Modeling language, SAWSDL (Semantic Annotations for WSDL and XML Schema). This interaction enables us to pair service functionalities through language understanding and reasoning. An optimisation problem is the another challenge with consideration in non-functional attributes, such as quality of service (QoS). Nowadays, many scholars have looked into optimisation problems in QoS-awareness web service composition applying AI planning and Evolution Computing techniques. However, it is not sufficient without considering good balancing between functional and nonfunctional quality in most scenarios. Therefore, the consideration in these two quality dimensions is a must, so that a comprehensive quality evaluation model is proposed to deal with finding optimised semantic web service composition solution. This paper develops a more applicable approach dealing with a comprehensive quality —semantic matchmaking and QoS optimisation for automated semantic web service composition.

1 Introduction

Service-oriented computing (SOC) is a novel computing paradigm that employs services as fundamental elements to achieve agile development of cost-efficient and integratable enterprise applications in heterogeneous environments [23]. Service Oriented Architecture (SOA) could abstractly realise service-oriented paradigm of computing, this accomplishment has been contributing to reuse of software components, from the concept of functions to units and from units to services during the development in SOA [5]. One of the most typical realisations of SOA is *web service*, which designated as "modular, self-describing, self-contained applications that are available on the Internet" [8]. Several standards play a significant role in registering, enquiring and grounding web services, such as UDDI, WSDL and SOAP [10].

Web service composition pertains to a combination of several web services to provide a value-added composite service that accommodates customers' arbitrarily complex requirements. This application is developed by integrating interoperable and collaborative functionalities over heterogeneous systems - diverse platforms and programming languages [7]. Due to an increase in a large-scale alignment of enterprise applications, the number of Web services has increased dramatically and unprecedentedly, which cause an immense redundancy in functionality in a huge searching space. Therefore, manual service and semi-automated web service composition are considered to be lower efficiency while automated web service composition is considered to be less human intervention, less time consumption, and high productivity.

Two most notable challenges for web service composition are ensuring interoperability of services and QoS optimisation [10]. Interoperability of web services is one challenge in the dimensions of syntactic and semantics [10]. The syntactic dimension has achieved success through the use of XML standards (such as $WSDL$, $SOAP$), the semantics dimension, on the other hand, demands further research. Adding semantics given by ontologies [21] enables web service to understand and reason with each other. Historically, there are many ontologies languages and formats for semantic service descriptions, such as OWL-S [18], WSML [11], and SAWSDL [14], The logical characteristics in reasoning makes "machine understanding" possible through identifying and matching semantic similarity in input/output parameters of web services in heterogeneous environments. The second challenge is to find optimised solutions to Quality of Service (*QoS*) over the composite web service. This problems give birth to *QoS-aware service composition* that considers the composition of service-level agreements [24] manifesting a collection of SLA rules and policies for supporting QoS-based composition.

Existing works on service composition are often addressed regarding one of the above challenges. One group optimises the quality of compositions under a pre-defined abstract workflow, which is considered to be a *semi-automated Web service composition* approach. Another group attempts to generate a composite plan automatically in discovering and selecting suitable web services, which are deemed to be an NP-hard problem [20]. *Semantic web services composition* is distinguished from the syntactic service composition. Its benefits are presented in eliminating conflicts at the semantic level of web service composition. In the last few years, substantial work has been done on semantic web service composition [10,15]. Howevever, few works have addressed an automatic semantic web service composition approach, which optimise both QoS and quality of semantic matchmatching as optimised web service composition sollutions. For example, customers prefer the highest quality of service solution associated with the acceptable semantic matchmatching quality.

The overall goal of this paper is to develop an comprehensive evaluation model in automated semantic web service composition that satisfactorily optimises both functional and non-functional requirements. Particularly, this project considers a semantic matchmaking quality extension to QoS awareness and a few structural constructs(sequence and parallel) for building optimised composition solution using Particle Swarm Optimisation. We aim to provide a more general and applicable way to measure semantic matchmaking in automated semantic web service composition and contribute to the state-of-the-art in this field. Three objectives of this work are accomplished as follows:

1. To address semantic matchmaking quality with considering different matching types associated with corresponding semantic similarity, which is demonstrated as a comprehensive quality — a more general and applicable evaluation method utilising semantic information of service descriptions in this paper.
2. To apply PSO for sorting an optimised service queue as a indirect representation of web service composition with consideration in optimising the comprehensive quality, this indirect representation is decoded using graph building algorithm to generate a service composite solution.
3. To evaluate the performance of proposed approaches, we utilise the datasets from Web Services Challenge 2009 (WSC'09) [13].

The remains of this paper are composed as the following: Sect. 2 provides the knowledge background on the semantic web service composition, including a literature review; Sect. 3 represents the an Introduction of of semantic automated web service composition approach, and the comprehensive quality evaluation model; Sect. 4 describes the experiments conducted to test the effectiveness of proposed model; Sect. 5 presents the results and analysis of these experiments; Sect. 6 is the paper conclusion.

2 Background

2.1 Problem Description

The purpose of web service composition is to accomplish an arbitrarily complex task fulfilling customer's requirement, which could be denoted as composite goal: $G(F(T_{Input}, T_{Output}), NF(T_{QoS}))$. This overall composite goal is demonstrated in two parts. One part is that a given input or input set to get the desired output or output set, it typically refers to a statement in functional requirement ; another part specify the overall acceptable composite quality, which covers aspects in non-functionality. To accomplish the first half goal, two stages are involved in the investigation of this paper : services discovery and service selection. Firstly, service discovery is to find matched web service: $S_n(F(S_{Input}, S_{Output}), NF(S_{QoS}))$ from Service Repository: $S\{S_1, S_2, ..., S_n\}$ with the given Task until the desired output found. It results in more than one solutions without considering optimisation; Service selection is to select web services to reach global best quality(T_{QoS}). We are aiming to optimise a combined objective function involving both functional and non-functional concerns. Particularly in this paper, two challenges mentioned in Sect. 1 are addressed in semantic web composite approach. First, the inputs of each service could be semantically matched by predecessor services in the composition. Second, optimisation in QoS would not be sufficient in many scenarios to make a good decision, if customers are looking for a trade off either in better service quality with counteroffer in QoS or better QoS with less strict functional matchmaking quality.

2.2 Semantic Web Service matchmaking Type

The semantic service matchmaking aims to discover appropriate services from service repository relevant to a customer's functional request defined by the Goal. A semantic web service is defined by $S(F(S_{Input} \in C_1, S_{Output} \in C_2), NF(S_{QoS}))$ with both Input and Output are linked to concept C_1 and C_2 in an ontology (O) respectively, satisfying $O = \{C, Tax\}$. The web service discovery process is to discovery the services an input-output concept matchmaking according to the Taxonomy(Tax) within an ontology (O). To measure the quality of semantic matchmaking, different matching levels that are typically considered in the literature [22]. We defined two web services associated with parameters in a particular domain: S_1 $(F(S_{Input} \in C_1, S_{Outputs} \in C_2), NF(S_{QoS}))$ and $S2$ $(F(S_{Input} \in C_3, S_{Output} \in C_4), NF(S_{QoS}))$ and an Ontology(O) with $C_1, C_2, C_3,$ and C_4 defined in.

- *Exact* ($\equiv$): Output of Web service S_1 and Input of Web service S_2 are Exact match ($S_{output} \in S_1 \equiv S_{input} S_2$), if Concept C_2 and Concept C_3 are equivalent.
- *Plugin* ($\sqsubseteq$): Output of Web service S_1 and Input of Web service S_2 are Plugin match ($S_{output} \in S_1 \sqsubseteq S_{input} \in S_2$), if Concept C_2 is a sub-concept of Concept C_3.
- *Subsume* ($\sqsupseteq$): Output of Web service S_1 and Input of Web service S_2 are Subsume matched($S_{output} \in S_1 \sqsupseteq S_{input} \in S_2$), if Concept C_2 is a sub-concept of Concept C_3.
- *Fail* ($\perp$). Output of Web service S_1 and Input of Web service S_2 are not matched (Fail) ($S_{output} \in S_1 \perp S_{input} \in S_2$), if none of the previous matches discovered.

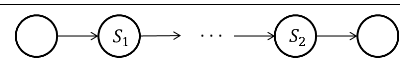
Note that, to find relevant services to automatically form suitable and applicable service compositions for a given goal, we consider Exact, Plugin and Fail match types in our paper. Therefore, these three robust and valid matches will be used to discover web service candidate for a service composition.

2.3 Quality of Service and Composition Constructs

Currently, most of the optimisation problems [9,12,17,27] in web service composition are focusing on QoS optimisation, which covers aspects in non-functional requirements. This problem have been explored in optimising both single objective and multi-objectives in QoS. Ideally, customers prefer lowest execution cost with highest response time and reliability. According to [32], four most often considered QoS parameters are as following:

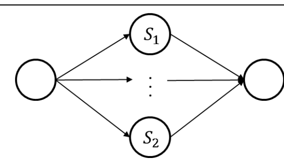
- *Response time* (T) measures the expected delay in seconds between the moment when a request is sent and the moment when the results are received.
- *Cost* (C) is the amount of money that a service requester has to pay for executing the web service
- *Reliability* (R) is the probability that a request is correctly responded within the maximum expected time frame.
- *Availability* (A) is the probability that a web service is accessible.

The aggregation value of QoS attributes for web services composition varies with respect to different constructs, which explains how services associated with each other [32]. Here we consider two composite constructs: sequence and parallel are used in building the composite flow, which are described as follows:



$$T = \sum_{n=1}^{m} t_n \quad C = \sum_{n=1}^{m} c_n \quad A = \prod_{n=1}^{m} a_n \quad R = \prod_{n=1}^{m} r_n$$

Fig. 1. Sequence construct and calculation of its QoS properties [31].



$$T = MAX\{t_n | n \in \{1, \dots, m\}\}$$

$$C = \sum_{n=1}^{m} c_n \qquad A = \prod_{n=1}^{m} a_n \qquad R = \prod_{n=1}^{m} r_n$$

Fig. 2. Parallel construct and calculation of its QoS properties [31].

Sequence construct The composite web service executes each atomic service associated with a sequence construct in a definite sequence order. The aggregation value for total time (T) and total cost (C) is the sum of time and cost of web services involved respectively. The overall availability and reliability in a sequence construct are calculated by multiplying their corresponding availability and reliability of each web service in probability theory. This construct is shown in Figure 1.

Parallel construct Web services in a parallel construct are executed in the meantime. The QoS aggregation value in total cost, availability and reliability are the same as these in sequence construct while the Total time (T) is delimited by the maximum time consumed path in the service composite solution. This construct is presented in Figure 2.

2.4 Related Work

Semantic web service matchmaking. Substantial work [2,19] on semantic web service composition focus particularly on functional requirements and negect non-functional requirements. This approach utilises Description Logic(DL) [1] reasoning between input and output concepts of web services to ensure a semantic matchmaking. Since semantic descriptions are introduced into web service, which is expressed in ontologies, the semantic matchmaking is to considered to seek similarity of parameters (i.e., input and output) of web services and pair a mapping between two knowledge representations encoded utilising the same ontology [16]. In [25], the author surveyed three approaches the similarity measures using taxonomies: one is based on nodes, in which similarity is determined by the information content of the nodes; one is entirely based on edge, where concept distance in a hierarchy structure is evaluated, and the hybrid approach that combines the previous two methods. A new similarity measure based on edge counting is introduced in a Ontology in [25], which extends similarity measure defined by Wu and Palmer [30]. Neighbourhood concepts are considered in their model in Formular 1, where $\lambda = 1$ (default value as 0).

$$q(s_{similarity})=\frac{2N \cdot e^{-\lambda L/D}}{N_1 + N_2} \tag{1}$$

In [15], web services are connected as an semantic links $sl_{i,j} \doteq \langle s_i, Sim_T(Out\_s_i, In\_s_j), s_j\rangle$, which apply one possible measure for similarity degree on nodes information demonstrated as Common Description rate of a semantic link in Formula 2. Extra Description $In_{s_x} \setminus Out_{s_y}$and Least Subsume $lcs(Out_{s_i}, In_{s_j})$. They chained the web service in five well-known matching types: Exact, Plug-in, Subsume, Intersection, and Disjoint. The quality of the semantic link is estimated as quality criteria q_m(i.e. either 1 (Exact), 0.75 (Plugin), 0.5 (Subsume) or 0.25 (Intersection)) associated with their corresponding quality of vector q_{cd} 2. The model is defined as follows.

$$q(sl_{i,j}) \doteq (q_{mt}(sl_{i,j}),\ q_{cd}(sl_{i,j})) \tag{2}$$

$$q_{cd}(sl_{i,j}) = =\frac{|lcs(Out_{s_i}, In_{s_j})|}{|In_{s_x} \setminus Out_{s_y} \dot{|} + |lcs(Out_{s_i}, In_{s_j})|} \tag{3}$$

However, the weakness of this approach for semantic link quality is that calculating Extra Description and Least Subsume of out-input request requires well and complete defined DL in knowledge representation in the different domains, which makes it nearly impossible to be widely applied to semantic matchmaking measures in today's' web services application. Additionally, in their semi-auto workflow approach, they do not consider data-flow compatible and automated service composition.

QoS-aware EC approaches. Evolution Computing techniques are considered to be more effective for solving globally optimisation problem. Genetic Programming (GP) approaches [29,28] are typical EC techniques widely applied to fully automated web service composition, which utilises a tree representation that service as the terminal nodes, and composite constructs as the functional nodes. The crossover and mutation process reproduce various individuals with ensuring correctness of structure. In [31], the author optimises the quality of the overall composition by a complex fitness function, which is also liable for correctness in functionality through penalising solutions. To simplify the checking of constraints for solutions, an indirect PSO-based approach was introduced in [28], this approach is to find a optimised web service queue in the way of searching best location using PSO for optimised QoS, that service queue is involved in building a graph representation of web service composition later. However, the neglecting of leveraging the semantic difference between composite service output and customers' request could lead to the lower quality of functionality matchmaking with best quality of service in QoS-awaren web service composite paper [9,12,17,27]. For example, a client asks for a particular AI textbook while output of the composte web service provides a general book. In this scenario, the output is a too broad consequences that have no meaning for the customers, even though web services are selected for a excellent composite QoS consideration. Therefore, we fill the gap to provide a more applicable approach to evaluate an overall quality in this paper.

3 QoS-aware Semantic Autoamted Web Serivce Composition

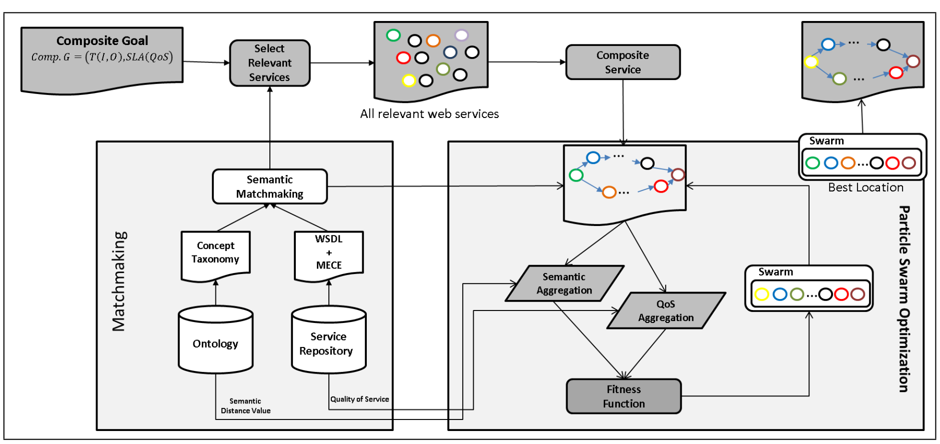


Fig. 3. Overview of the proposed approach.

On the support of the representation of our solution, in this section, we introduce our evaluation model in a graph-based representation for QoS-aware automatic semantic web service composition utilising PSO, which is considered to be simple and efficient without penalising and repairing comparing to GP [27]. Fig. 3 shows the overview of our approach with the several steps included. The composite process is triggered by a composite goal defined in Subection 2.1, which demonstrate customers requirements in terms of functional and non-functional requirement. The previous one is well defined as an overall input-output requirement, and the later one is the users' accepted quality of service. Those requirements are initially used to discover all relevant web services and their relationship in semantic matchmaking to get prepared for building graph representations. The second step of selecting relevant web services contributes to shrink the size of participated web services for dimensions of particles involved in future optimised problem. In the third step, a service composition graph randomly built up with considering the QoS and matchmaking quality. This graph building is interleaved with semantic matchmaking for services. The output of one service and its associated input of another service is calculated in the matchmaking phase where semantic matchmaking quality are measured which will be explained in Subsection 3.3. The further optimization of service composition solution is performed over the PSO in Section 3.1, which is used to find optimised particle locations mapped back to web service for service-graph building. This approach [28] is an indirect representation of web service composition that constructing a graph-based composite service composition.

3.1 QoS-aware Semantic Web Service Composition algorithm

The overall algorithm investigated here is a combination of a simple forward decoding PSO [28] and a graph-building technique [4] with our comprehensive quality evaluation model. The idea of that is to translate the particle location into a service queue as an indirect representation of service composition graph, finding the best fitness of the composite service solution is to figure out the optimised location of the particle. In PSO, the dimension of the particle is set up as the same as the number of relevant web services shrunk from serivce repository, and all of them are mapped to the particle location, put in a service queue and sorted in ascending order. The second method is to decode a corresponding graph built from that service queue. and it is a forward graph-building technique from a start node to the end node, we select and connect web services to startNode if this web service can satisfy the given Task input initialised in a Output Set, and this Output Set is updated after adding web service to the graph, later on, we services are keeping being selected and added if its added services' outputs in the Output Set could be satisfied, if the any outputs in the Output Set meets the Task's requriment, we end up graph building. In addition, dangle services and edges shoud be removed. At last, fitness value can be calculated by the construct-based aggregation.

Algorithm 1. Steps of graph-based PSO optimisation technique.

1. Randomly initialise each particle in the swarm.
while *max. iterations not met* **do**
  forall *particles in the swarm* **do**
    2. Sort partcile's postion vector as a service queue in ascending order
    3. Initialise OutputSet with Task Input
    4. Initialise a graph with a startNode associated with Task Input as its output
    while *max. OutputSet not contains Task Input* **do**
      5. Get web service from service queue
      if *All inputs of current web service satisfied with output from outputSet* **then**
        6. Create and connect service edge associated with matchmaking quality to graph.
        7. Create and connect node(current web serivce) to graph.
        8. Remove current web service fromt the queue.
        9. Update OutputSet with outputs current web service.
      10. Get next web service from service queue
    11. Connect endNode the graph
    12. Remove dangle services from the graph
    13. Remove dangle edges
    14. Aggregation of semantic matchmaking and QoS
    15. Calculate the particle's fitness.
      $fitness_i = w_1 SDst + w_2 A_i + w_3 R_i + w_4(1 - T_i) + w_5(1 - C_i)$
      where $\sum_{i=1}^{5} w_i = 1$
      if *fitness value better than pBest* **then**
        16. Assign current fitness as new pBest.
      else
        17. Keep previous pBest.
  18. Assign best particle's pBest value to gBest, if better than gBest.
  19. Calculate the velocity of each particle according to the equation:
    $v_{id} = v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id})$
  19. Update the position of each particle according to the equation:
    $x_{id} = x_{id} + v_{id}$

3.2 Semantic matchmaking

The semantic matchmaking is achieved by utilising semantic annotation with OWL and MECE ontologies [3] with OWL and XWL format respectively. In OWL, the semantics specifies the both concepts and individuals in forms of classes and instances but limited to taxonomies. The semantic concepts are expressed with the their statements and relationship in RDFs (such as subClassOf), and the semantic individuals are related to the input and output parameters of web services. The MECE is an extensive annotation to WSDL files, which defines the each parameter of services associated with their semantic individuals. The overall matchingmaking process is demonstrated in Figure 4.

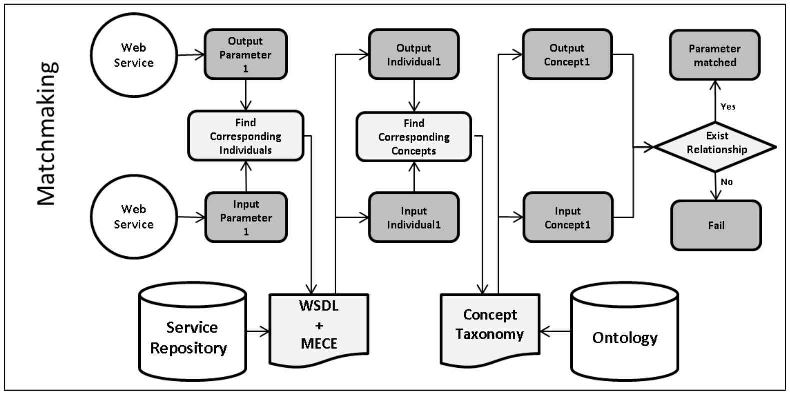


Fig. 4. Semantic matchmaking.

The semantic matchmaking transfer a function match (OutputS1, InputS2) to a pair of concept match (C2, C3) for the two web services defined in Section 2.2. The matching mechanism attempts to determine whether correct match criteria between the source concepts of C1 and the target concepts of C2, and the quality of semantic matchmaking for the two its reasoning is also calculated in the quality model in the following Section refqulityModel. The advantage of utilising semantic matchmaking is to enable a reasonable connection between two web services from different service providers to find more service solutions with better QoS.

3.3 Comprehensive quality model and aggregation matrix

In this paper, we proposed a semantic matchmaking model to evaluate the semantic quality of two matched web services. Then we fill the gap of current prevailing QoS-aware optimisation consideration in the fully automated composition approach, which has not considered the quality of matchmaking for service selection.

Semantic matchmaking model. Due to the discretizational characteristics of different match types, the match type quality is driven by the cost of data type integration and manipulation [15], so that match type is considered to be one factor for the semantic quality. Another factor in our proposed model is concept similarity, which could be evaluate based on edge counting method defined by [25], a taxonomy based measurement for ontology matching, In our paper, it could be used to estimate the parameter concepts similarity between the output and input request for selecting the highest semantic-closeness web services. Therefore, given the quality match type and the similarity of two parameters, the semantic matchmaking quality is defined as Equation 4.$q(s_{matchType})$ is set up as the same as the quality of the semantic link in literature [15] (i.e. either 1 (Exact), 0.75 (Plugin), 0.5 (Subsume) or 0.25 (Intersection)), and $q(s_{similarity})$ is Equation 1.

$$q_{smq} \doteq (q(s_{matchType}),\ q(s_{similarity})) \tag{4}$$

Comprehensive quality model. Compared to QoS evaluation model, the comprehensive quality model is established to investigate both functional and non-functional attributes of composite web service. The non-functional properties typically refer to availability, reliability, cost and response time, which are given by service providers. Consequently, the comprehensive quality model integrated the semantic matchmaking quality and quality of service in Equation 5, which could be broken down into Equation 6.

$$q_{cq} \doteq (q(s_{smq}),\ q(s_{QoS})) \tag{5}$$

$$q_{cq} \doteq (q(s_{mt}),\ q(s_s),\ q(s_a),\ q(s_r),\ q(s_c),\ q(s_t)) \tag{6}$$

Quality aggregate matrix.The quality aggregation is defined based on the constructs of composite web services on the criteria of functional and non-functional properties. Typically, two web services are connected by more than one parameter-related concepts. In details, edge-level quality aggregation is determined by the mean of parameter-related concept quality in both semantic match type and similarity respectively, and construct-level quality is further calculated based on the edege-level quality aggregation following the rules in Table 1, which is the quality aggregate matrix for semantic web service composition. The calculation of QoS aggregation is similar to [6] while semantic matchmaking aggregation is adopted from [15].

Table 1. Quality aggregate matrix for semanic web service composition

Composition Construct			Sequence	Parallel
QualityFactors	Functional	$Q(s_{mt})$	$\prod_{n=1}^{m} q(s_{mt})$	$\prod_{n=1}^{m} q(s_{mt})$
		$Q(s_s)$	$(\sum_{n=1}^{m} q(s_s))/m$	$(\sum_{n=1}^{m} q(s_s))/m$
	NonFunctional	$Q(s_a)$	$\prod_{n=1}^{m} q(s_a)$	$\prod_{n=1}^{m} q(s_a)$
		$Q(s_r)$	$\prod_{n=1}^{m} q(s_r)$	$\prod_{n=1}^{m} q(s_r)$
		$Q(s_c)$	$\sum_{n=1}^{m} q(s_c)$	$\sum_{n=1}^{m} q(s_c)$
		$Q(s_t)$	$\sum_{n=1}^{m} q(s_t)$	$max(q(s_t))$

3.4 Fitness Calculation

The fitness value is calculated from the aggregate value of quality components involved in a graph-based service composition. A weighted sum of those components is utilised in the fitness function 7, in which fitness value of 1 means the best overall quality and 0 means the worst. In addtion, aggregate value of each parameters must be normalised between 0 and 1. In details, MT, S, A, and R are calculated in Formula 8 while T and C are calculated in Formula 9.

$$fitness_i = w_1MT + w_2S + w_3A_i + w_4R_i + w_5(1 - T_i) + w_6(1 - C_i) \tag{7}$$

where $\sum_{i=1}^{6} w_i = 1$

$$normalise(MT, S, A, orR) = \begin{cases} \frac{value-min}{max-min} & \text{if } max - min \neq 0. \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

$$normalise(TorC) = \begin{cases} \frac{max-value}{max-min} & \text{if } max - min \neq 0. \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

4 Experiment Design

In this section, a quantitative evaluation approach is adopted in our experiment design. The objective of the evaluation are: 1) measure the effectiveness of the proposed comprehensive quality model in automated semantic web service composition approach, and explore the impact of the semantic matchmaking that contributes to overall composition quality. 2) compare the optimised sollutions with using comprehensive evaluation model with using QoS-aware evaluation model to reveal the advantages of our novel method.

We utilise the web service challenge 2009 (WSC09) web service data sets to perform a standard evaluation. Compared to previous web service challenge dataset, WSC'09 is evolved with OWL for providing richer semantics. With an incremental number in concept, individuals and web services in each dataset, it is a proper dataset for the scalability measures in our defined quality evaluation model.Table 1 presents the features of the WSC09 dataset. The number of concepts, individuals in the ontology and services of each data set is shown in the second, third, fourth column respectively. Also, we extend the all the datasets with QoS attributes from service providers to perform a comprehensive quality evaluation.

Table 2. features of the WSC'09 datasets

Dataset	No.Concept	No.Individual	No.Service
WSC'09 01	1578	3102	572
WSC'09 02	12388	24815	4129
WSC'09 03	18573	37316	8138
WSC'09 04	18673	37324	8301

We run the experiment under computing grid comprising of almost 170 NetBSD (Unix operating system) workstations operated by the Sun Grid Engine. Experimentation was done using the composite service approach explained in Section 3, and evaluated in the comprehensive quality model introduced in Section 3.3. The parameters were chosen based on general settings from [26], In PSO approach, 30 particles evolved in 100 generations in the searching space, in which problem dimension's size equals to the relevant service size shown in graph. we run 30 times independently for each dataset. In addition, weights setting are more flexible defined by users' preferences. Simplely, we configure weights of fitness function as same leverage for balancing functional side and nonfunctional side. Therefore, W1 and W2 are equals to 0.1 and 0.4 entirety 0.5, and W3, W4, W5, W6 are all set to 0.125 summed 0.5.

5 Results and Analysis

5.1 Convergence Test

The purpose of our experiment is to expose the results of our proposed approach are optimised comprehensive quality for web service challenge tasks utilising datasets from WSC09. To study an overall effectiveness of evaluation model and investigate the impacts of all involved components that are contributing the weighted sum fitness functional. We analysis all the performance of these components in the whole evolutionary process in Figure 5, in which the three tasks' experiment results are arranged in four groups consisting of average fitness, average matchType quality, average similarity quality and average quality of service for generation 0-99 with optimum.

Firstly, the average fitness value over 100 generations is calculated by the average best fitness so far found per generation over 30 independent runs. The marked value plotted in the first group of the figure ranges from 0 to 1, and it is considered to be more optimised if its value is closer to value 1. We can see that there is a significant increase in the fitness value between generation 0 and generation 15-25, the remained generation continues to perform a very slight improvement in the fitness value and reach a plateau with no further changes. Furthermore, it is a common feature in three tasks average fitness graph. Also, there is a fast convergence of the population can be observed. That is simply because of many repetitive solutions as the indirect web service composition representation, although it is a large search space.

We also investigate the variation in functional quality part including two subfactors — average matchType quality and average similarity quality from generation 0 to 99 in second and third groups in Figure 5. In these groups, both the average matchType quality and average similarity quality are the average value corresponding to the average best fitness so far found, and further calculated their mean over 30 independent runs respectively. The dominant tendency of the marked value representing a similar characteristic compared to the average fitness value group trend. However, there are some slight fluctuations over the generations, and it could be reasonably treated as a typical situation that theoretically single objective function prevents selecting the lowest semantic matchmaking quality without granting for the excellent, yet the combination of all the influential functional quality part evolved towards an optimised solution.

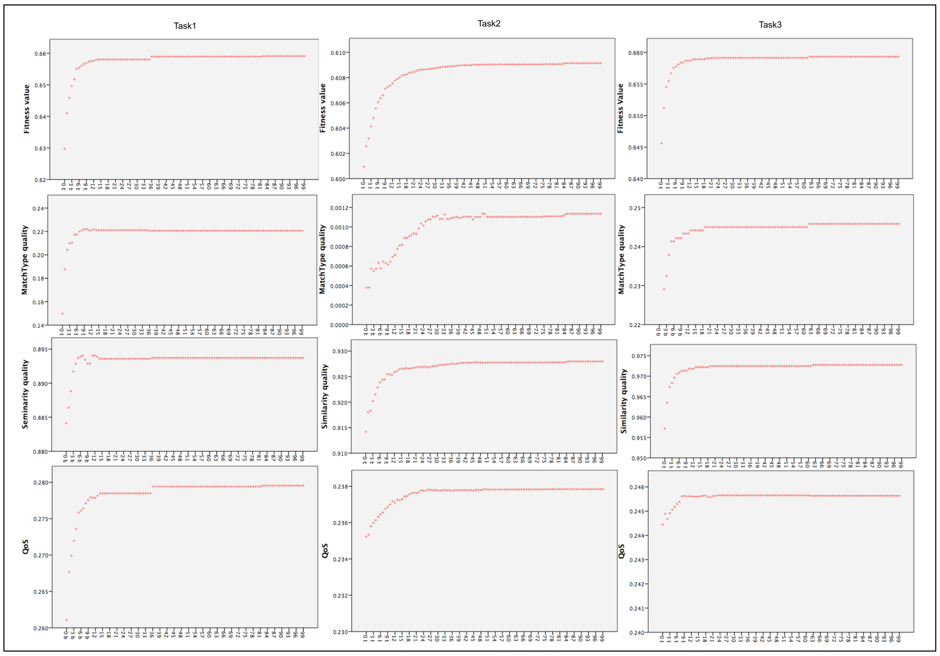


Fig. 5. Average fitness, Average MatchType quality, Average Similarity quality and Average QoS per generation with comprehensive quality optimum

Lastly, It is interesting to look at the average QoS per generation for the 30 independent runs if the behaviour of functional part performance is towards a higher value, where QoS is the value corresponding to the best fitness so far found in each generation. We construct the results in the fourth groups in Figure 5. It is obviously that the overall trend of QoS moves upward in three tasks, and reach almost constant later on. Luckiely, we reach a very good state for QoS value, because we don't see too much trade-off from QoS regarding the improvement in functional quality part.

5.2 Comparision Test

We defined our optimised web service composition as $WSC = (V, E)Graph$, where V is a set of services as vertex: $V = [S1, S2...Sn]$ and E is a set of service connections as edges associated matchType quality and semantic similarity quality: $E = e_1, e_2, ...e_n$ where $e1$ is define as $(S_1, S_2) = q_{mt}, q_s$. Here we provide an uncomplicated example of the web service composition solution to problem1, which can be described in the Figure 6. The data of composite web service flows from the start to the end, where five web services involved and linked with each other using edge connections. Besides that, q_{mt} and q_s) are calcuated for all edges $e_1, e_2, ...e_n$, and further aggregateed in Q_{mt} and Q_s

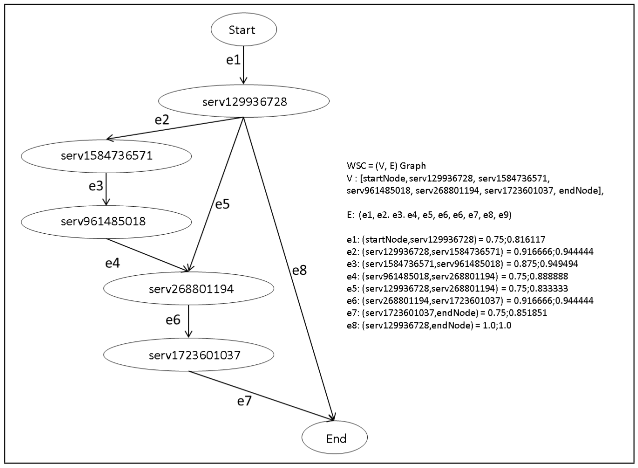


Fig. 6. Example composition solutions for problem1 WSC'09.

respectively.In addtion, Q_{QoS} is calcuated the all the non-functional attributes of V using aggregation rules defined before.

After well-defined solutions are discussed, we will look at Q_{mt}, Q_s and Q_{QoS} using black-box testing for sollution comparision, where these components in both comprehensive quality evaluation approach and traditional QoS-ware one [9,12,17,27] are compared to reveal how good our evaluation mode is. In a decision table 3, these two models are interpreted as a rule set considering three different conditions (three independent tasks) associated with three corresponding sollutions. In each sollution, we compare Q_{mt},Q_s and Q_{QoS}, where Q_{QoS} is normalised to make them comparable from 0 to 1 in our method. From the table below.

From the table 3, Task 1 represents the same solution that can be observed in values of Q_{mt},Q_s and Q_{QoS} using the smallest dataset size with the lowest complexity of ontologies and valid solutions. However, we can recognise trade-offs in task2 and task3 using our evaluation model, where a 0.021668 increase and a 0.00098 decrease in similarity quality and matchType quality with weights 0.4 and 0.1 respectively so that it is still considered to be an improvement in functional quality part with a trade-off value 0.007249 in QoS. In task3, the benefit of functional part quality is clearer with an increase in both similarity quality and matchType quality of 0.056648 and 0.078862 while a slight decrease in QoS. In conclusion, we can perceive that our evaluation could find out better functional quality with a reasonable trade off in QoS.

6 Conclusion

This work introduced a more applicable and simpler evaluation model for semantic automated web service composition that relies on semantic matchmaking

Table 3. Quality of two evaluation model in a decision table

		Rule 1 QoS-aware Evaluation	Rule 2 Comprehensive Quality Evaluation
Condition	dataset1	√	√
	dataset2	√	√
	dataset3	√	√
Task1 Optimised sollution	Q_{mt}	.232635	.232635
	Q_s	.903572	.903572
	Q_{QoS}	.578577	.578577
Task2 Optimised sollution	Q_{mt}	.001828	.000851↓
	Q_s	.917172	.938840↑
	Q_{QoS}	.480251	.473002↓
Task3 Optimised sollution	Q_{mt}	.191188	.247836↑
	Q_s	.951509	.973723↑
	Q_{QoS}	.493927	.491538↓

quality extension to QoS-aware consideration. The key idea of these evaluation model is to consider the quality of semantic matchmaking corresponding to different matchmaking types, which is simply achieved by applying the similarity measures in parameter-related concepts, and those web services are involved in further selection for dealing optimised problems. Also, the effectiveness of our model is also proved to be able to obtain a better-optimised solution in better functional quality with a reasonable trade-off in QoS. Future works in this area should investigate in multi-objective optimisation as well as consider improving efficiency in the functional quality calculation.

References

1. Baader, F.: The description logic handbook: Theory, implementation and applications. Cambridge university press (2003)
2. Bansal, S., Bansal, A., Gupta, G., Blake, M.B.: Generalized semantic web service composition. Service Oriented Computing and Applications 10(2), 111–133 (2016)
3. Bleul, S., Comes, D., Kirchhoff, M., Zapf, M.: Self-integration of web services in bpel processes. In: Proceedings of the Workshop Selbstorganisierence, Adaptive, Kontextsensitive verteilte Systeme (SAKS) (2008)
4. Blum, A.L., Furst, M.L.: Fast planning through planning graph analysis. Artificial intelligence 90(1), 281–300 (1997)
5. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web services architecture. w3c working note. W3C Working Notes (2004)
6. Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web 1(3), 281–308 (2004)
7. Casati, F., Georgakopoulos, D., Shang, M.C.: Technologies for E-Services: Second International Workshop, TES 2001, Rome, Italy, September 14-15, 2001. Proceedings, vol. 2. Springer Science & Business Media (2001)

8. Curbera, F., Nagy, W., Weerawarana, S.: Web services: Why and how. In: Workshop on Object-Oriented Web Services-OOPSLA. vol. 2001 (2001)
9. Feng, Y., Ngan, L.D., Kanagasabai, R.: Dynamic service composition with service-dependent qos attributes. In: Web Services (ICWS), 2013 IEEE 20th International Conference on. pp. 10–17. IEEE (2013)
10. Fensel, D., Facca, F.M., Simperl, E., Toma, I.: Semantic web services. Springer Science & Business Media (2011)
11. Fensel, D., Lausen, H., Polleres, A., De Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: Enabling semantic web services: the web service modeling ontology. Springer Science & Business Media (2006)
12. Huang, Z., Jiang, W., Hu, S., Liu, Z.: Effective pruning algorithm for qos-aware service composition. In: 2009 IEEE Conference on Commerce and Enterprise Computing. pp. 519–522. IEEE (2009)
13. Kona, S., Bansal, A., Blake, M.B., Bleul, S., Weise, T.: Wsc-2009: a quality of service-oriented web services challenge. In: 2009 IEEE Conference on Commerce and Enterprise Computing. pp. 487–490. IEEE (2009)
14. Lausen, H., Farrell, J.: Semantic annotations for wsdl and xml schema. W3C recommendation, W3C pp. 749–758 (2007)
15. Lécué, F.: Optimizing qos-aware semantic web service composition. In: International Semantic Web Conference. pp. 375–391. Springer (2009)
16. Lécué, F., Léger, A.: A formal model for semantic web service composition. In: International Semantic Web Conference. pp. 385–398. Springer (2006)
17. Ma, H., Wang, A., Zhang, M.: A hybrid approach using genetic programming and greedy search for qos-aware web service composition. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XVIII, pp. 180–205. Springer (2015)
18. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., et al.: Owl-s: Semantic markup for web services. W3C member submission 22, 2007–04 (2004)
19. Mier, P.R., Pedrinaci, C., Lama, M., Mucientes, M.: An integrated semantic web service discovery and composition framework (2015)
20. Moghaddam, M., Davis, J.G.: Service selection in web service composition: A comparative review of existing approaches. In: Web Services Foundations, pp. 321–346. Springer (2014)
21. O'Leary, D.: Review: Ontologies: A silver bullet for knowledge management and electronic commerce. The Computer Journal 48(4), 498–498 (2005)
22. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic matching of web services capabilities. In: International Semantic Web Conference. pp. 333–347. Springer (2002)
23. Papazoglou, M.P.: Service-oriented computing: Concepts, characteristics and directions. In: Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on. pp. 3–12. IEEE (2003)
24. Sahai, A., Machiraju, V., Sayal, M., Van Moorsel, A., Casati, F.: Automated sla monitoring for web services. In: International Workshop on Distributed Systems: Operations and Management. pp. 28–41. Springer (2002)
25. Shet, K., Acharya, U.D., et al.: A new similarity measure for taxonomy based on edge counting. arXiv preprint arXiv:1211.4709 (2012)
26. Shi, Y., et al.: Particle swarm optimization: developments, applications and resources. In: evolutionary computation, 2001. Proceedings of the 2001 Congress on. vol. 1, pp. 81–86. IEEE (2001)

27. da Silva, A.S., Ma, H., Zhang, M.: A graph-based particle swarm optimisation approach to qos-aware web service composition and selection. In: 2014 IEEE Congress on Evolutionary Computation (CEC). pp. 3127–3134. IEEE (2014)
28. da Silva, A.S., Mei, Y., Ma, H., Zhang, M.: Particle swarm optimisation with sequence-like indirect representation for web service composition. In: European Conference on Evolutionary Computation in Combinatorial Optimization. pp. 202–218. Springer (2016)
29. da Silva, A., Ma, H., Zhang, M.: Graphevol: A graph evolution technique for web service composition. In: Chen, Q., Hameurlain, A., Toumani, F., Wagner, R., Decker, H. (eds.) Database and Expert Systems Applications, Lecture Notes in Computer Science, vol. 9262, pp. 134–142. Springer International Publishing (2015)
30. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. pp. 133–138. Association for Computational Linguistics (1994)
31. Yu, Y., Ma, H., Zhang, M.: An adaptive genetic programming approach to qos-aware web services composition. In: 2013 IEEE Congress on Evolutionary Computation. pp. 1740–1747. IEEE (2013)
32. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: Proceedings of the 12th international conference on World Wide Web. pp. 411–421. ACM (2003)