Composition Optimizer: A Tool for Optimizing Quality of Semantic Web Service Composition*

Freddy Lécué
The University of Manchester
Booth Street East
Manchester, UK
+44 161 306 8929
firstname.lastname@manchester.ac.uk

ABSTRACT

Ranking and optimization of web service compositions are some of the most interesting challenges at present. Since web services can be enhanced with formal semantic descriptions, forming the "semantic web services", it becomes conceivable to exploit the quality of semantic links between services (of any composition) as one of the optimization criteria. For this we propose to use the semantic similarities between output and input parameters of web services. Coupling this with other criteria such as quality of service (QoS) allow us to rank and optimize compositions achieving the same goal. We present the Composition Optimizer tool, using an innovative and extensible optimization model designed to balance semantic fit (or functional quality) with non-functional QoS metrics, in order to optimize service composition. To allow the use of this model in the context of a large number of services as foreseen by the EC-funded project SOA4All we propose and test the use of Genetic Algorithms.

Keywords

Web service, semantic Web, service composition.

1. INTRODUCTION

The *Composition Optimizer* prototype[1] [5] focuses on optimizing web service composition, where services are composed on the basis of the semantic similarities between input and output parameters as indicators of service functionality. To measure semantic similarity, we use the concept of (functional) semantic link [7], defined as a semantic connection (i.e., part of data flow) between an output and an input parameter of two services. Web service compositions could thus be estimated and ranked not only along well known non functional parameters such as *Quality of Services* (QoS) [2] but also along the dimension of semantic similarity as indicator of functional fit [6]. In this work the *Composition Optimizer* tool computes the composition by optimising both the non functional QoS and the quality of semantic fit along non-trivial data flow, where the information required and provided by services does not match perfectly in every data flow, using semantic-based description of services. By addressing non trivial data flow in composition, we aimed at limiting the costs of (semantic heterogeneity) data integration between services by considering appropriate quality of semantic links.

*Foundation Project: Supported by European Commission VII Framework IP Project Soa4All.

[1]http://tinyurl.com/ndaqdz

2. OUR APPROACH

The *Composition Optimizer* maximizes the quality of service composition by resolving a multi-objective optimization problem with constraints on quality of services and semantic links (which is known to be NP-hard). Most approaches addressing optimization in web service composition are based on *Integer linear Programming* (IP) e.g., [8]. However, IP approaches have been shown to have poor scalability [2] in terms of time taken to compute optimal compositions when the size of the initial set of services grows. Such a case can arise in the future semantic web, where a large number of services will be accessible globally. This is the vision of SOA4All, a strategic EC-funded project. Rapid computation of optimal compositions is especially important for interactive systems providing service composition facilities for end users, where long delays may be unacceptable.

Here we demonstrate that the optimisation problem can be automated in a more scalable manner using *Genetic Algorithms* (GAs), and propose an approach to tackle QoS-aware *semantic* web service composition. In more details we reuse the notion of semantic link (and their common descriptions) between web services. Therefore our prototype focuses on the quality criteria for QoS-aware semantic web service composition. To this end, GA-based evolutionary approach has been suggested, including the strategies of the crossover, mutation and fitness function.

3. CONTEXT AND IMPLEMENTATION

Semantic web services are defined by their semantic descriptions using an $\mathcal{ALE}$ ontology (formally defined by 1100 concepts and 390 properties, 1753 individuals) in Telecommunication Domain, provided by a commercial partner. We have incorporated random values for Qos parameters (price and response time). Common description rate and matching quality, as quality criteria of semantic links, have been computed according to an on-line DL (Description Logic) reasoning process.

The common description rate is calculated by computing the *Extra Description* [6], the Least Common Subsumer [1], and the size ([4] p.17) of DL-based concepts. These DL inferences and the matching types have been achieved by a DL reasoning process i.e., an adaptation of Fact++ [3] for considering DL difference. Our GA is implemented in Java, extending a GPL library[2]. The optimal compositions are

[2]http://jgap.sourceforge.net/

computed by using an elitist GA where the best 2 compositions were kept alive across generations, with a crossover probability of 0.7, a mutation probability of 0.1, a population of 200 compositions. The roulette wheel selection has been adopted as selection mechanism. We consider a simple stopping criterion i.e., up to 400 generations. We conducted experiments on Intel(R) Core(TM)2 CPU, 2.4GHz with 2GB RAM.

4. EXPERIMENTAL RESULTS

Besides computing the optimal composition, we analyze our approach by i) evaluating performance after decoupling the GA and the (on-line) DL reasoning processes, and ii) comparing the convergence of our approach with [2].

4.1 GA Process vs. DL Reasoning

Since our approach is mainly depending on DL reasoning (i.e., Subsumption, Difference and *lcs* for semantic links quality) and the GA-based optimization process, we decouple and detail the computation costs of our approach regarding compositions of with up to 30 tasks and 35 candidates per task (35^2 candidate semantic links between 2 tasks).

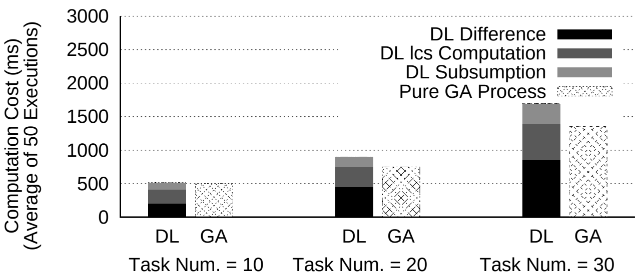

Figure 1: DL and GA Processes in our Approach.

DL reasoning is the most time consuming process in optimisation of QoS-aware semantic web service composition wherein the number of tasks and candidate services are greater than 10 and 35. This is caused by the critical complexity of q_{cd} computation through DL Difference (even in $\mathcal{ALE}$ DL).

4.2 Convergence of GA-Based Approaches

We compare the convergence of our approach with the main alternative at present [2]. To this end the functional criteria of our approach are disregarded in order to focus only on the GA-driven aspects of the optimisation process. According to Table 1, the advantage of our approach is twofold. Firstly we obtain better fitness values for the optimal composition than the approach of [2]. Secondly, our approach converges faster than the approach of [2]. In addition our function avoids getting trapped by local optimums by i) further penalizing compositions that disobey constraints and ii) suggesting a dynamic penalty, i.e., a penalty having a weight that increases with the number of generations.

These results support the adoption of our model in the cases where a large number of tasks and services are considered.

5. RELATED TOOLS

Review of existing tools to optimize web service compositions reveals that no tool has specifically addressed optimisation of service composition using both *QoS* and *semantic similarities* dimensions in a context of *significant scale*. Indeed main tools focus on either QoS [2, 8] or on functional criteria such as semantic similarities [6] between output and input parameters of web services for optimising web service composition. In contrast, we present an innovative model that addresses both types of quality criteria as a trade-off between data flow and non functional quality for optimizing web service composition.

Tasks Num.	Approach	Max. Fitness (%)	Generation Num.	Time (ms)
10	Our Model	99	120	1012
	[2]	97	156	1356
20	Our Model	97	280	1650
	[2]	94	425	2896
30	Our Model	95	360	3142
	[2]	85	596	6590

Table 1: Comparing GA-based Approaches (Population size of 200).

We follow [2] and suggest the use of GAs to achieve optimization in web service composition, yet we also extend their model by i) using semantic links to consider data flow in composition, ii) considering not only QoS but also semantic quality (and constraints) of composition, iii) revisiting the fitness function to avoid local optimal solution (i.e., compositions disobeying constraints are considered).

6. CONCLUSION

We have presented *Composition Optimizer* i.e., a tool that supports *QoS*-aware *semantic* web service composition in a context of significant scale. On the one hand the benefits of a significant domain such as the Web is clear e.g., supporting a large number of services providers, considering large number of services that have same goals. On the other hand, the benefits of combining semantic links between services and QoS are the optimisation of data flow quality and the non functional quality of compositions.

7. REFERENCES

[1] F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. In *DL*, 2004.

[2] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani. An approach for qos-aware service composition based on genetic algorithms. In *GECCO*, pages 1069–1075, 2005.

[3] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *KR*, pages 636–649, 1998.

[4] R. Küsters. *Non-Standard Inferences in Description Logics*, volume 2100. Springer, 2001.

[5] F. Lécué. Optimizing qos-aware semantic web service composition. In *ISWC*, pages 375–391, 2009.

[6] F. Lécué, A. Delteil, and A. Léger. Optimizing causal link based web service composition. In *ECAI*, pages 45–49, 2008.

[7] F. Lécué and A. Léger. A formal model for semantic web service composition. In *ISWC*, pages 385–398, 2006.

[8] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. Quality driven web services composition. In *WWW*, pages 411–421, 2003.