Quality of Service based service selection approaches in web service composition

Ondrej Balun
Student of master program
TU Wien
Austria
ondrejbalun@gmail.com

Dominik Kertys
Student of master program
TU Wien
Austria
dominik.kertys@gmail.com

Christoph Paur
Student of master program
TU Wien
Austria
cpaur@gmx.at

ABSTRACT

Web services have become an important part of modern technology spectrum. Services can be composed into various web service compositions. There are two crucial aspects on the web service composition - functional and non-functional requirements. Functional requirements focus on the functionality of the composed service, e.g., given an auction service, an example of functional requirements is that a bid lower than 10€ will be automatic rejected. Non-functional requirements are concerned with the Quality of Service (QoS), e.g., an example of the auction service non-functional requirements is that the service will respond to each new bid within 2 seconds. Non-functional requirements are important to web service composition, because they are often a clause in Service-level Agreements (SLAs). Even though the functional requirements are satisfied, a non-secure or unreliable service cannot be used. In our paper we provide a brief overview and comparison to the existing approaches how to select appropriate service for service composition in order to QoS.

Categories and Subject Descriptors

H.3.5 [**On-line Information Services**]: Web-based services; D.3.4 [**Systems and Software**]: Distributed systems

General Terms

Quality of service, Service-level agreement, Performance, Measurement

Keywords

Web service composition, service selection, quality of service, service-level agreement

1. INTRODUCTION

Web services and associated Quality of Service (QoS) have gained more attention in research community with growing number of complex systems based on Web service composition (WSC), which are called composite services. WSC allows binding heterogeneous applications and interoperability between companies. Building good QoS is a strategic task in Service selection - the problem of selecting the most appropriate service from the set of existing ones that best match required functional and non-functional requirements. Especially in the Business-to-Business (B2B) orchestration, Service-level Agreement (SLA) need to be defined as the contract between service provider and service requestor clarifying the measurements and constraints for non-functional requirements (e.g. performance, availability, trust, security) and also penalty if SLA will be broken. For non-specific clients are SLAs reduced to best effort delivery.

2. RELATED WORK

Zeng et al. (2003) Their selection is based on two approaches one is using a local selection for process and global selection of services. In 2004 was this model reused adding semantics into data, so the selection the best candidate was made with the help of semantics.[6]

Yu and Lin (2005) Proposed an approach for selecting the service as a Multiple-Choice Knapsack Problem (MCKP). Services processes were divided into two groups: 1. with sequential flow structure, 2. with general flow structure including loops, conditionals and parallel operations. To solve the MCKP in each group they used heuristic algorithms and made many simulations and comparisons.[7] They also implemented framework where QoS is evaluated and framework selects the web services which met user request. Time duration, execution rate, reputation, execution cost and availability as QoS attributes was considered.

Feng et al. (2007) Consider multiple QoS constraints and found the utility based function for each QoS parameter. They have considered both the functional and non-functional behavior of the QoS parameters and created a service class which consists of many candidate services with different non-functional parameters but with common functionality. They proposed an architecture which has composer to decompose user request and QoS analyzer to get QoS requirements from database and find the corresponding service from UDDI.[3] This standard web service models were in 2012 extended to explicitly support WSC. The architecture consists of three components: business process, a composition process and a description process.

3. QUALITY CRITERIA FOR COMPOSITE SERVICES

Table (Tab. 1) describes the most important QoS criteria and their computation for composite services execution plan p.

Criteria	Aggregation function
Price	$\sum_i q_{pr}(s_i, op(t_i))$
Duration	$CPA(p, q_{du})$
Reputation	$\frac{1}{N}\sum_i^N q_{rep}(s_i)$
Success rate	$\prod_i (q_{rat}(s_i)^{z_i})$
Availability	$\prod_i (q_{av}(s_i)^{z_i})$

Table 1: Aggregation functions for computing the QoS

Execution price
Sum of execution prices over operations invoked by services s in the execution plan. $op(t_i)$ denotes operation invoked by task t_i.
Execution duration
For an operation of a service, the execution duration measures the expected delay in seconds between the moment when a request is sent and the moment when the results are received. In composite services is the execution duration for execution plan computed using Critical Path Algorithm (CPA). Execution plan is decomposed into oriented weighted digraph (Fig. 1). The critical path in the graph has the biggest sum of execution durations of the invoked service operations. Execution durations are the weights in the graph.

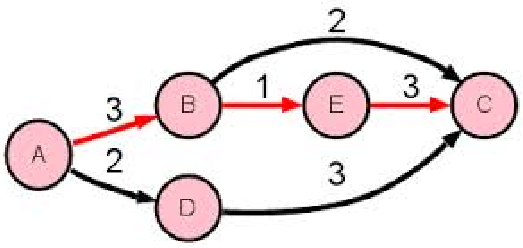

Figure 1: Weighted digraph with critical path.

Reputation
Reputation means the trustworthiness often dependent on end-user experiences. In service composition is computed as the average of reputations all participating services.
Successful execution rate
Can be computed as the products of the factor of successful rates all services in execution plan, where z_i is set to 1 if the service is critical, otherwise is set to 0 with the assumption that if non-critical service fails, it will not delay the whole composite service execution.
Availability
Availability of the execution plan is defined in the same way as Successful execution rate. Again, we assume that, when the non-critical service is unavailable, a service can be reselected without delaying the whole composite service execution.

4. SERVICE SELECTION

The service selection is the very important step in web service composition process. Available services are divided into classes with respect to their functionality. Services with the same functionality are in the same class, so input to service selection is set of classes or groups. Each service has a Service Profile where provider offers values for QoS attributes to compete among other services in the same class. In order to [5] there are three categories of such that QoS attributes:

Technical domain independent, e.g. response time, availability, robustness, reliability

Non-technical domain independent, e.g. execution price, penalty, discount, reputation

Domain dependent meaningful in specific domain, e.g. refresh time

4.1 Optimization-Based Approaches

Local optimization

The idea of local optimization is to select one process/service from each group of candidates independently on the other groups. Using a given utility function, the values of the different QoS criteria are mapped to a single utility value and the process/service with maximum utility value is selected. This approach is very efficient in terms of computation time as the time complexity of the local optimization approach is $O(l)$, where l is the number of service candidates in each group.[1] The offered value for the j_{th} QoS attribute, q_j, ($j \in J$: set of all QoS attributes), by web service s, is mapped to a value between 0 and 1 using a single attribute linear utility function U_j in Eq. 1. In this equation, q_j^{max} and q_j^{min} are the maximum and minimum values offered for q_j by all the candidate web services of the same functionality group.[5]

$$U_j(q_j) = \begin{cases} \frac{q_j - q_j^{max}}{q_j^{max} - q_j^{min}} & \text{if larger } q_j \text{more desirable} \\ \frac{q_j^{max} - q_j}{q_j^{max} - q_j^{min}} & \text{if smaller } q_j \text{more desirable} \end{cases} \quad (1)$$

Local optimization is mostly used for decentralized environments but it is not suitable in case if global constraints (end-to-end) are required, because these cannot be evaluated locally.

Global optimization

Local optimization solves the local selection problem but it may not lead to global optimality (end-to-end) because of global constraint (e.g. maximum total price). Global optimization tries to solve the problem on the composite service level. The aggregated QoS values of all possible service combinations are computed, and the service combination that maximizes the aggregated utility value, while satisfying global constraints is selected.[1] The global selection problem can be modeled as a Multi-Choice Multidimensional Knapsack problem (MMKP)[1], which is known to be NP-hard in the strong sense. To solve this problem Integer Linear Programming - for small models, or genetic algorithms - for more complex models, are used. One possible solution for linear programming is linear minimization/maximization. If for example the service requester wants to minimize the price

[1]http://www.es.ele.tue.nl/pareto/mmkp/

U_p and maximize the availability U_a, he can use maximization process (Eq. 2) of weighted QoS attributes (weights w_{Qos} can represent the importance of each attribute). Price should be minimal, so for maximization process reversed value for the price can be used ($U_p{}' = 1 - U_p$)[5].
Minimize:

$$w_P \cdot U_P + w_A \cdot U_A \quad (2)$$

Subject to:

$$U_P = \sum_i \frac{P^{MAX} - \sum_j p_{ij} \cdot x_{ij}}{P^{MAX} - P^{MIN}} \quad (3)$$

$$U_A = \sum_i \frac{\prod_j a_{ij} \cdot x_{ij} - A^{MIN}}{P^{MAX} - P^{MIN}} \quad (4)$$

Where p_{ij} is the price of service x_{ij} for executing task j, then $\sum_j p_{ij} \cdot x_{ij}$ is the sum of all offers for the tasks. Minimum and maximum price offers for the composition are denoted as P^{MAX} and P^{MIN}.

4.2 Negotiation-Based Approaches

Negotiation in context service selection is an automated process done by negotiator such as agent, web service or third-party broker. Negotiation is used for automatic SLA creation. In this process negotiator negotiate over SLA terms to come up with formal SLA at the end of process.[5] Two ways of negotiation are available by the assumption that service provider is already chosen (determined) or not. **Per-contractual SLA Negotiation:** one-to-one negotiation, the service provider is determined. The goal is to negotiate service parameters to define concrete service with concrete non-functional requirements.

Dynamic provider selection: one-to-many process after service discovery, used for service provider selection that best matches the requestor requirements

Automated negotiation should contain three critical components:[4]

Negotiation Protocols - communication rules govern the message exchange, include permissible types of negotiator; negotiation states; events, which lead to state change; a valid actions for all states

Negotiation Issues - the range of issues over which agreement must be reached. At one extreme, the object may contain a single issue (such as price), while on the other hand it may cover hundreds of issues (related to price, quality, timings, penalties, terms and conditions, etc.).

Decision-making Model - the rules to decide when to start negotiation, what is acceptable range, how to prepare offer

4.3 Hybrid Approach Optimization + Configuration Approach

Work by Comuzzi and Pernici [2]. In their approach, rather than providing a single value for each QoS attribute, the service provider publishes the set of values that they can support for each QoS attribute. The selection process follows this three phases:

Phase 1: Matchmaking. When a request is received, the first phase filters the providers of services in the requested category, by retaining only those that satisfy, at least partially, the requestors' QoS requirements and budget constraints.

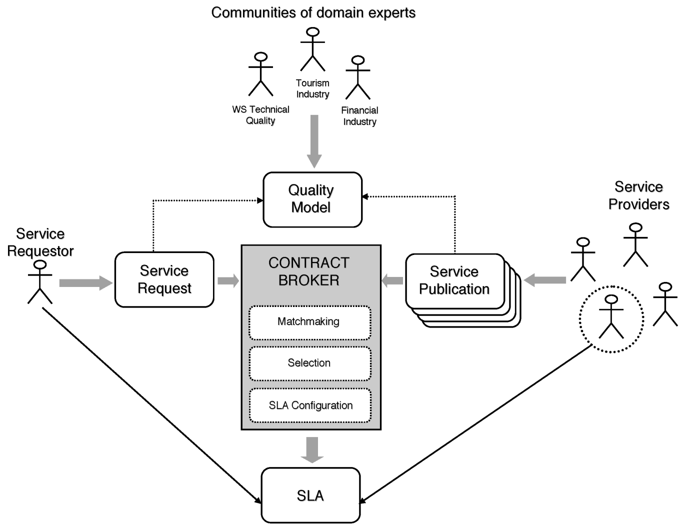

Figure 2: The Web service contracting framework.

Phase 2: Provider Selection. One provider, among the providers filtered in the previous phase, is selected. Specifically, providers are ranked by a bidding function and the provider of the service offer associated to the lowest bid is selected.

Phase 3: Agreement Configuration. The contract broker applies the requestors' strategy to configure the selected provider's service offer in order to obtain the closed SLA.

5. COMPARISON

The optimization approaches, focused on optimizations models with constraints and decision variables, are restricted to situations where QoS Profile of the service is more fixed and pre-determined. This is the main disadvantage in the dynamic process of web service composition. But because of pre-determined QoS Profile these technics are easier to use and maintain. More realistic approaches are the automated negotiation which are agent-based and use the negotiation model with negotiation objects and issues, negotiation protocol and coordination. These are putted mostly in the situations, where the price should be negotiated. In the optimizations have been applied mathematical optimization techniques in comparison to the economics, coordination and game theory techniques, which are linked with automated negotiation. There are common known solvers and genetic algorithms for optimization strategy but for negotiation there are only some domain-specific tools.

6. CONCLUSION AND FUTURE WORK

As the number of web services with similar functionality increases, the QoS criteria are become more competitive. In our paper we presented the most common approaches how to select the best services for composite service that match requestors' demand. We provided a brief comparison of the two main popular service selection approaches. The future research search for better heuristics and genetic algorithms for optimization process and tries to find more complex and multi-disciplinary model to support automated negotiation.

7. REFERENCES

[1] M. Alrifai and T. Risse. Combining global optimization with local selection for efficient qos-aware service composition. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 881–890, New York, NY, USA, 2009. ACM.

[2] M. Comuzzi and B. Pernici. A framework for qos-based web service contracting. *ACM Trans. Web*, 3(3):10:1–10:52, July 2009.

[3] X. Feng, Y. Ren, J. Hu, Q. Wu, and Y. Jia. A model for service composition with multiple qos constraints. In *Computing: Theory and Applications, 2007. ICCTA '07. International Conference on*, pages 208–213, 2007.

[4] N. Jennings, P. Faratin, A. Lomuscio, S. Parsons, M. Wooldridge, and C. Sierra. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

[5] M. Moghaddam and J. Davis. Service selection in web service composition: A comparative review of existing approaches. In A. Bouguettaya, Q. Z. Sheng, and F. Daniel, editors, *Web Services Foundations*, pages 321–346. Springer New York, 2014.

[6] M. Rajeswari, G. Sambasivam, N. Balaji, M. S. Basha, T. Vengattaraman, and P. Dhavachelvan. Appraisal and analysis on various web service composition approaches based on qos factors. *Journal of King Saud University - Computer and Information Sciences*, 26(1):143 – 152, 2014.

[7] T. Yu, Y. Zhang, and K.-J. Lin. Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Trans. Web*, 1(1), May 2007.