



HAL
open science

Web Service Composition with Uncertain Non-functional Parameters

Łukasz Falas, Pawel Stelmach

► **To cite this version:**

Łukasz Falas, Pawel Stelmach. Web Service Composition with Uncertain Non-functional Parameters. 4th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Apr 2013, Costa de Caparica, Portugal. pp.45-52, 10.1007/978-3-642-37291-9_5. hal-01348734

HAL Id: hal-01348734

<https://hal.science/hal-01348734>

Submitted on 25 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Web Service Composition with Uncertain Non-Functional Parameters

Łukasz Falas, Paweł Stelmach¹

¹Wrocław University of Technology, Institute of Computer Science, Wybrzeże
Wyspiańskiego 27, 50-370 Wrocław, Poland
{Lukasz.Falas, Pawel.Stelmach}@pwr.wroc.pl

Abstract. The problem of service composition is of key importance in the Internet of Things paradigm. A composite Web Service can clearly determine how real life objects described as Web Services could interact with each other. This article proposes a non-functional parameters aggregation algorithm, used in composite service plan optimisation stage, taking into account the uncertainty of the non-functional parameters. Also, an optimisation algorithm, using the proposed aggregation method, is presented to show the complete solution for the problem. In this proposition, the algorithm focuses on composite service plan optimisation with uncertain parameters with the given uncertainty description on example of uncertainty parameter. However, the proposed algorithm can be generalized, so it can be used also with other uncertain parameters.

Keywords: QoS-oriented service composition, uncertain non-functional parameters, Service Oriented Architecture.

1 Introduction

In the Internet of Things paradigm real life objects can be described with Web Services. Web Services are often considered building blocks for composite services performing complex tasks. The same approach could be used in the Internet of Things paradigm to show how Internet users can interact with a composition of objects, which is more than simply a sum of its parts. This composability aspect of Web Services enables us to fulfil user requirements, which cannot be met by any single available Web Service. The service composition task in general aims to fulfil both functional and non-functional user requirements (the latter are known also as Quality of Service parameters). This paper focuses on the QoS-oriented composition and defines the problem as composite service plan optimisation with uncertain non-functional parameters for each Web Service.

The aspect of uncertainty of non-functional parameters is often omitted in the general service composition problem, however it is crucial to ensuring that the Web Service model is properly describing the real behaviour of the Web Service. In real life scenarios Web Services exist in dynamically changing environments and their parameters – like execution time or cost – vary, depending on other parameters

values. Therefore Web Services, as real life objects, cannot be described with pre-determined non-functional parameters values.

The subject of Web Services composition is often raised in literature. In various papers approaches like AI Planning [1][2], workflow based service composition[3] or template based service composition[4][5] were suggested. Those approaches mainly focus on satisfying the functional requirements for the service. With time more and more authors point that the semantics of composition, user requirements and of Web Services are not regarded sufficiently. Many take on the topic of QoS-Aware service composition focusing on fulfilment of non-functional requirements [6][7].

However in the QoS-Aware service composition approach the problem of non-functional parameters uncertainty was often omitted. Papers where non-functional parameters uncertainty is only a side-topic to the general service QoS optimisation problem often use a simple probabilistic approach based on mean calculation for uncertain parameters, which is later used as a fixed value of given non-functional parameters[8]. Papers, which focus on the problem of non-functional parameters uncertainty, also utilize other probabilistic measures (i.e. skewness or curtosis) [9][10] to achieve reliable results. Some of them even propose a comprehensive solution for the problem of non-functional parameters uncertainty [11].

2 Relationship to Internet of Things

Service Oriented Architecture (SOA) is an approach, in which some functionality of a web application can be provided as Web Services. Nowadays Web Services are still mostly useful to programmers, offering general operations that each can provide actions on multiple objects. In the future however, it is more likely that a single Web Service could be responsible for a single real-life object, like a light in a house or a TV. The latter is already possible but not popular as programmers still think about more general methods that help them handle multiple problems with a single snippet of code. However, this creates a demand for more monolithic environments, which stands in opposition to the SOA paradigm. It is true that operations (like “show device state”) will be available via Web Services and accessible remotely from different locations, thus making the system distributed, but single objects like TV, coffee express, fridge or washing machine will be only data in a database in a single centralized management system. In contrast, each device could be available via a Web Service and managed independently of another by various management systems. In fact, as in the Internet of Things paradigm, we could manage them in a direct manner, looking at Web Services as wrappings for real-life objects.

In this context we cannot use a typical approach when a Web Service could be executed hundreds of times at the same time until the server can't process any more requests. Here, the “execution time” of the object-based Web Service is not equal to duration of single use of that object but should rather be described in general (for instance with uncertainty), considering the demand on that object as usually two clients cannot use one Web Service at the same time.

3 Formal Problem Definition

3.1 Service Composition

In general, service composition process consists of two steps. First, the required functionalities and their interactions – i.e. control and data flow – are identified. Second, for a set of the functionalities appropriate candidate services are discovered from the repository and then selected in an optimisation task resulting in a composite service execution plan describing a required composite service.

In this paper we will not describe a detailed approach on how candidate services are discovered and assume that a composite service execution plan with service candidate for each step of the plan is available. In further sections we will describe an algorithm for composite service plan optimisation, in which from sets of candidates for each step of the plan only one Web Service is selected in such a manner that the quality of the whole composite service, determined by QoS parameters, meets the non-functional requirements.

A composite service plan, denoted by S , is described with a graph, in which nodes denote steps of the plan, which are in fact users functional requirements. We denote them by:

φ_i – i -th functionality (functional requirement, node in a graph)

Candidate services are services that meet the functional requirements but differ in non-functional parameters. For each functionality φ_i there are m_i candidate services denoted by:

s_{ij} – j -th candidate service for functionality φ_i

In the process of composite service plan optimisation, for each functionality φ_i , only one Web Service will be selected from a corresponding set of candidate services.

The selection is in general based on non-functional parameters of each candidate, (referred to as Quality-of-Service – QoS attributes) like service execution time, service execution cost, service availability, service execution success rate, service reputation or service execution frequency. The quality of the composite service is determined by the quality of selected services, meaning that each services' QoS property contributes to the QoS calculated for the whole composite service. Depending on the composite service structure (determined by the graph that represents the service plan) the non-functional parameters of the composite service will be calculated differently, with respect to the functional parameter type. For example *cost* of a composite service could be a simple sum of a cost of each selected service, whereas *service execution time* is a sum of execution times whenever services are executed one after another (in a series) and a maximum when services are executed in parallel.

Non-functional parameters of a composite service are denoted by $A_k(S)$ to highlight that they are calculated in the process of aggregation of QoS parameters of services selected in the plan S , where k is an index determining for which non-functional parameter of the QoS parameters vector ($k = 1$: execution cost, $k = 2$: execution time, $k = 3$: availability, $k = 4$: success rate, $k = 5$: reputation, $k = 6$: execution frequency), the aggregation is calculated.

Additionally a constraint (C_k) can be defined for each non-functional parameter:

$$\begin{cases} A_k(S) \leq C_k & \text{if } k = 1,2 \text{ (time, cost)} \\ A_k(S) \geq C_k & \text{if } k = 3,4,5,6 \text{ (availability, etc.)} \end{cases}$$

Finally, the quality criterion is defined as follows:

$$Q(S) = \min \left[\left\{ \sum_{k=1}^2 w_k \max \left(\frac{A_k(S) - C_k}{C_k}, 0 \right) \right\} + \left\{ \sum_{k=3}^6 w_k \max \left(\frac{C_k - A_k(S)}{C_k}, 0 \right) \right\} \right]$$

where w_k is a weight of the k -th QoS constraint defined by the user ($k = 1,2, \dots, 6$)

3.2 Uncertainty

As a base for our uncertainty model we use uncertain variable definition introduced by Bubnicki in [12]. In this approach one cannot define a certain value for a specific parameter but instead an expert defines a confidence index for a set of values, expressing his confidence that those values could be obtained. In general, the expert can define a function h that assigns parameters values to appropriate confidence index values.

Typically, one would ask the expert if the value of a parameter belongs to some set D and then with function h , a certainty of that statement could be determined. However, in presented case we want to ask the opposite question. Assuming a certain degree of confidence (value for confidence index) we would like to determine what are the most certain values for the parameter.

To be exact, in service composition, we would like to know what is the uncertain availability of a given candidate service s_{ij} , assuming that its confidence index is greater than $c(s_{ij}) \stackrel{\text{def}}{=} c_{ij}$. Taking into consideration a certain demand d for the service s_{ij} and confidence index c_{ij} (stating that the demand was in the same timeframe), the approximated value of the k -th non-functional parameter of service s_{ij} is denoted as:

$$\bar{q}_{ij}^k(c_{ij}, d) = [q_{ij,s}^k, q_{ij,e}^k]$$

In the following sections we focus on the uncertainty coming from uncertain demand for a service and we give an example of uncertainty based on availability.

4 Research Contribution and Innovation

4.1 Service Plan Optimisation Algorithm

The service plan optimisation algorithm presented in this paper is based on a combined approach using simulated annealing and tabu search. The general frame of the algorithm is as follows: first greedily find an initial plan, then if it doesn't meet the requirements search for a neighbour plan. Innovations presented in our approach are situated especially in determining values for uncertain non-functional parameters,

in determining locally the best candidate service for a given functionality with a use of a fitness function (which is used to generate both initial and neighbour plans) and in so called QoS aggregation (used to verify the quality of the generated plan: both initial and neighbour). Limitations of this paper did not allow presenting all algorithms of the approach, focusing only on those algorithm improvements relevant to the uncertainty aspect of service composition.

4.2 Determining Uncertainty in Non-Functional Service Parameters

With varying demand for a specific service the function h , defining what certainty can be attributed to which parameters values, will also be different (this is depicted in Fig.1), thus we use denotation h_d to indicate that the function h is determined for a specific demand d .

The confidence index c_{ij} is determined based on the certainty that requests for the service s_{ij} were made in the same timeframe, thus it is greatly influenced by the timeframe in which the demand for a service was calculated.

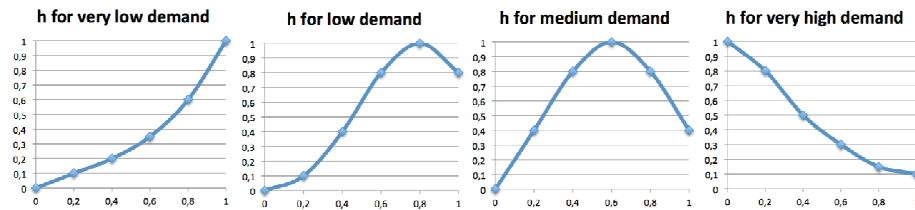


Fig.1. Shape of function h (assigning confidence index to parameter values) for varying service availability depending on demand on a service.

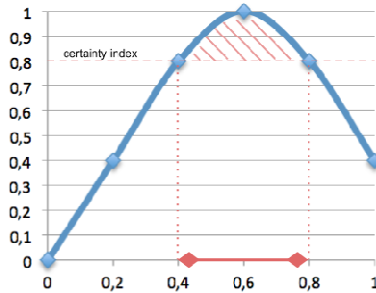


Fig. 2. Method for determining the uncertain value (of availability) assuming a certainty index

For instance, if there was only 1 request then, with appropriate function h_1 , we could assume that current confidence index would be equal to 1 (only 1 service request was made in that time so we are sure that the timeframe is “the same”) and we would know the exact availability. Were there d requests, we would use function h_n and assuming the level of certainty c_{ij} (that those requests were made in the same timeframe) we could calculate the uncertain value of availability parameter (also see Fig.2):

$$\bar{q}_{ij}^k(c_{ij}, d) = q_{ij}^k * \arg h_d(c_{ij}) = [q_{ij,s}^k, q_{ij,e}^k]$$

The above is true assuming that function h_d is dependent only on popularity d and not varying for different services.

q_{ij}^k is a typical value of k -th non-functional parameter for a service s_{ij} if no uncertainty is present.

4.3 Atomic Service Fitness Function

The goal of a fitness function is to establish a local fitness score according to weighted QoS requirements and thus to define a single order on candidate services (for a given functionality) to jointly represent their various non-functional parameters. In initial execution plan generation algorithm this function is used directly for each functionality to determine greedily which candidate service is locally the best.

Input: q_{ij}^k - value of the k -th QoS parameter of service s_{ij} ; $\{q_{ij}^k | 1 \leq i \leq m_i\}$ - values of the k -th QoS parameter for candidate services for functional requirement φ_i (there are m_i of those services), set of candidate services SC

Output: Aggregated QoS (fitness) estimate for the s_{ij} service

```

1.  for each service  $s_{ij}$  in SC do
2.      for  $k=1$  to 6 do
3.

$$q_{ij}^k = \begin{cases} -\frac{q_{ij}^k - \text{mean of } \{q_{ij}^k | 1 \leq i \leq m_i\}}{\text{std deviation of } \{q_{ij}^k | 1 \leq i \leq m_i\}} & \text{for } k = 1, 2 \text{ (time, cost)} \\ \left[ \frac{q_{ij}^k - \text{mean of } \{q_{ij}^k | 1 \leq i \leq m_i\}}{\text{std deviation of } \{q_{ij}^k | 1 \leq i \leq m_i\}}, \frac{q_{ij}^k - \text{mean of } \{q_{ij}^k | 1 \leq i \leq m_i\}}{\text{std deviation of } \{q_{ij}^k | 1 \leq i \leq m_i\}} \right] & \text{for } k = 3 \text{ (availability)} \\ \frac{q_{ij}^k - \text{mean of } \{q_{ij}^k | 1 \leq i \leq m_i\}}{\text{std deviation of } \{q_{ij}^k | 1 \leq i \leq m_i\}} & \text{for } k = 4, 5, 6 \text{ (other)} \end{cases}$$

4.      end
5.      end
6.  return  $[w_3 q_{ij}^k + \sum_{k \in \{1, 2, 4, 5, 6\}} w_k q_{ij}^k, w_3 q_{ij}^k + \sum_{k \in \{1, 2, 4, 5, 6\}} w_k q_{ij}^k]$ 

```

In the above algorithm availability is a non-functional parameter with uncertain values, thus its normalization and summation with other parameters is not a typical sum but instead the fitness score for the atomic service is also an uncertain number.

Then, depending on user approach (optimistic, pessimistic, average) appropriate order on candidate services can be imposed (based on maximum, minimum or average value).

4.4 Service Plan Aggregation Algorithm

This aggregation algorithm recursively establishes a vector of non-functional parameters for any composite service provided with a service execution plan S with QoS parameters for each of atomic services in that plan.

The algorithm replaces sequence and parallel service structures with single services whose QoS parameters are aggregated so that the new aggregated value is calculated according to formulas in Table 1 (where $q^k(\varphi_i)$ is a value of k -th non-functional parameter of a service currently selected for functionality φ_i). The general concept for such a Table is known in literature for composite service plan optimisation. Here,

formulas for availability aggregation take into account the uncertain nature of this non-functionality.

| Param. | Cost | Execution Time | Availability (uncertain) | Success rate | Reputation | Execution Frequency |
|------------|-------------------------------|-------------------------------|--|--------------------------------|---|---|
| Sequential | $\sum_{i=1}^n q^1(\varphi_i)$ | $\sum_{i=1}^n q^2(\varphi_i)$ | $[\prod_{i=1}^i q_s^3(\varphi_i), \prod_{i=1}^i q_e^3(\varphi_i)]$ | $\prod_{i=1}^i q^4(\varphi_i)$ | $\frac{1}{n} \sum_{i=1}^n q^5(\varphi_i)$ | $\frac{1}{n} \sum_{i=1}^n q^6(\varphi_i)$ |
| Parallel | $\max q^1(\varphi_i)$ | $\max q^2(\varphi_i)$ | $[\min q_s^3(\varphi_i), \min q_e^3(\varphi_i)]$ | $\min q^4(\varphi_i)$ | $\frac{1}{n} \sum_{i=1}^n q^5(\varphi_i)$ | $\frac{1}{n} \sum_{i=1}^n q^6(\varphi_i)$ |

Table 1. QoS aggregation formulas for various non-functional parameter types.

5 Discussion of Results

The algorithms presented in previous sections allow for composite service optimisation, considering that some parameters are described with uncertain values. Regarding limitations of this paper only availability parameter was described in more detail, as its dependency on users demand on a service is most clear. Other non-functional parameters like execution time or cost could also be presented as uncertain parameters and the main approach to optimisation wouldn't have to be changed, however more in depth discussion would be needed. For instance uncertain execution time leads to significantly different h function shapes compared to availability parameter, depending on demand on that service. With each request for a certain service in a specific timeframe the uncertainty of expected experienced execution time (considered as waiting time and actual execution time) increases significantly, ranging from t to $d*t$, where d is the demand and t is the typical execution time.

6 Conclusions and Further Work

In this paper a service composition problem in the Internet of Things paradigm was discussed. We presented an algorithm for composite service plan optimisation with uncertain non-functional parameters. In the presented works some parts of the algorithm were presented in more detail, namely a method for calculating uncertain availability parameter value, estimation of fitness function for service candidates with uncertain parameter and a method for aggregation of non-functional parameters of services considering series and parallel composition structures.

Algorithms presented in this paper will be included in the PlaTel platform for composition and management of telecommunication services [13] and are a step in an on-going research on composite service composition and its application to various domains like telecommunication sports or transport. Further testing of the algorithms will be performed in the PlaTel framework, using an advanced testing framework for Web Services currently in development.

Further work will focus on uncertainty description of other non-functional parameters besides availability. Finally, more research will be done in the matter of demand for a service at a specific timeframe, how that timeframe is established and how it can influence uncertainty of parameters in other timeframes.

Acknowledgments. The research presented in this paper has been co-financed by the European Union as part of the European Social Fund.

References

1. Xin, L., Xinhuai, T., Zhaoteng, S., Xiaozhou, Y., Delai, C.: AFlow: An Automated Web Services composition system based on the AI planning and workflow. In: Proceedings 2010 International Conference on Progress in Informatics and Computing, 1067- 1071 (2010)
2. Wu, Z., Ranabahu, A., Gomadam, K., Sheth, A. P., Miller, J. A.: Automatic Composition of Semantic Web Services using Process and Data Mediation. In: Proceedings of the 9th Intl. Conf. on Enterprise Information Systems (2007)
3. Pathak, J., Lutz, S. B. R., Honavar, V.: MoSCoE: A Framework for Modeling Web Service Composition and Execution. In: 22nd International Conference on Data Engineering Workshops, IEEE Computer Society (2006)
4. Sirin, E., Parsia, B., Hendler, J.: Template-based Composition of Semantic Web Services. In: AAAI Fall Symposium on Agents and the Semantic Web, Virginia, USA (2005)
5. Jie, G., Bo, C., Junliang, C., Lei, Z.: A Template-Based Orchestration Framework for Hybrid Services. In: 2008 Fourth Advanced International Conference on Telecommunications, Athens, Greece. p. 315-320 (2008)
6. Ko, J. M., Kim, C. O., Kwon, I.-H.: Quality-of-service oriented web service composition algorithm and planning architecture. In: Journal of Systems and Software, vol. 81, no.11, pp. 2079-2090 (2008)
7. Yu, T., Zhang, Y., Lin, K.-J.: Efficient algorithms for Web services selection with end-to-end QoS constraints. In: ACM Transactions on the Web, vol.1 no.1, p.6-es (2007)
8. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. In: IEEE Transactions on Software Engineering., vol. 30, no. 5, pp. 311-327(2004)
9. Wan, C., Wang, H.: Uncertainty-aware QoS Description and Selection Model for Web Services. In: Proceedings of the 2007 International Conference on Services Computing, pp. 154-161 (2007)
10. Wiesemann, W., Hochreiter, R., Kuhn, D.: A Stochastic Programming Approach for QoS-Aware Service Composition. In: Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid, Lyon, France (2008)
11. Hwang, S.Y., Wang, H., Tang, J., Srivastava, J.: A probabilistic approach to modeling and estimating the QoS of web-services-based workflows. In: Information Sciences 177, 5484–5503 (2007)
12. Bubnicki, Z.: Uncertain variables in the computer aided analysis of uncertain systems. In: Lecture Notes in Computer Science, Springer-Verlag, vol. 1798, s. 528-542 (2000)
13. Świątek, P., Stelmach, P., Juszczyszyn, K., Prusiewicz, A.: Service Composition in Knowledge-based SOA Systems. In: New Generation Computing 30.2: 165-188 (2012).