Dieter Schuller, André Miede, Julian Eckert, Ulrich Lampe, Apostolos Papageorgiou, Ralf Steinmetz: *QoS-based Optimization of Service Compositions for Complex Workflows*. In: Proceedings of the Eighth International Conference on Service Oriented Computing, Springer, San Francisco, USA, December 2010, pp. 641-648. ISBN: 978-3-642-17357-8.

QoS-based Optimization of Service Compositions for Complex Workflows

Dieter Schuller, André Miede, Julian Eckert, Ulrich Lampe, Apostolos Papageorgiou, and Ralf Steinmetz

Multimedia Communications Lab (KOM),
Technische Universität Darmstadt, Germany
firstname.lastname@KOM.tu-darmstadt.de

Abstract. In Service-oriented Architectures, business processes can be realized by composing loosely coupled services. If services in the Internet of Services with comparable functionalities but varying quality levels are available at different costs on service marketplaces, service requesters can decide, which services from which service providers to select. The work at hand addresses computing an optimal solution to this service-selection-problem considering complex workflow patterns. For this, a linear optimization problem is formulated, which can be solved by applying integer linear programming techniques.

Keywords: Optimization; Service Selection; Quality of Service; Complex Workflows

1 Introduction

In highly competitive markets with similar products and services, enterprises are facing a tough cost pressure. Their offered products and services need to be constructed and provided efficiently. Therefore, efficient process execution is mandatory. With respect to the globalization and the deregulation of markets, enterprises are forced to react quickly to changing environments, driven by market forces, and adapt their business processes. This requires the business processes to be flexible. But as IT architectures within an enterprise are often heterogeneous, the required flexibility is hard to achieve. Even within a single enterprise, a certain amount of legacy systems and a couple of applications exist running on different operating systems and middleware platforms, implemented with different programming languages. An approach for integrating these legacy systems and applications is necessary in order to realize the required flexible and efficient business processes.

In Service-oriented Architectures (SOA), business processes can be realized by composing loosely coupled services, which autonomously provide a more or less complex functionality depending on their granularity (cf. [1]). To support and enable agile business processes, the SOA paradigm is often recommended [2]. In the Internet of Services, typically multiple service providers offer equal or

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

rather similar services regarding the services' functionalities at different Quality of Service (QoS) levels and at different costs on service marketplaces. This gives enterprises the opportunity to select those services which meet their business and QoS requirements best. The problem of selecting appropriate services which meet specified QoS and cost conditions and restrictions for business processes – the service-selection-problem – is well known in the literature and has been discussed recently by several authors [3–5]. An optimal selection of services and their invocation results in an efficient business process execution. Therefore, the work at hand addresses the computation of an optimal solution to the service-selection-problem for complex workflows by formulating a linear optimization problem, which can be solved optimally by applying integer linear programming (ILP) [6]. It extends our approach in [7] by taking recursive interlacings of these patterns into account.

The rest of this work is structured as follows. In Section 2, we distinguish our approach from related work in this field. The considered workflow patterns and the applied system model is depicted in Section 3. An approach for recursively interlacing the required QoS aggregation functions is discussed in Section 4. In Section 5, the computation of an optimal execution plan taking recursive workflow pattern interlacings into account is addressed. Finally, in Section 6, conclusions are drawn and future work is discussed.

2 Related Work

A lot of work has been done regarding the service-selection-problem. Several authors propose and implement heuristic solutions to this problem [3,8,9]. Further, also tree-based algorithms are proposed in [4]. Here, the complex workflow is transformed into a tree-like execution structure (BPTree) based on the alignment of the considered workflow patterns. An optimal solution to the service-selection-problem is proposed in [10, 11]. The respective authors solve this problem for complex workflows by unfolding arbitrary cycles and computing optimal execution plans for each possible path in the complex workflow, thereby formulating and solving a linear optimization problem using ILP solely for sequential workflows.

This is overcome in the work at hand, as (in our approach) the knowledge of all possible execution paths is not necessary in advance. Our aim is to describe and implement an approach that enables the computation of an optimal solution to the service-selection-problem for complex workflows without considering each execution path separately and for recursive interlacings of these patterns, which – to the best of our knowledge – has not been addressed in the literature so far. Thus, we create a sort of benchmark, enabling other (heuristic) approaches to evaluate their achieved solution quality without having to calculate the optimal solution with brute force algorithms (checking out the whole solution space) but by (simply) applying ILP. In case, scalability issues are foregrounded, the heuristic solution method proposed in [12] can be applied to our approach.

3 System Model

In order to formulate the mentioned linear optimization problem – consisting of an objective function and a set of process conditions – in Section 5, we present the applied system model in this section. Thereby, an abstract business process (written, e.g., in Business Process Modeling Notation – BPMN) consisting of n abstract process steps respectively tasks (we will use these terms synonymously) is assumed. In the work at hand, the process steps may be arraged according to the workflow patterns *sequence*, *parallel split* (AND-split), *synchronization* (AND-join), *exclusive choice* (XOR-split), *simple merge* (XOR-join), which are described in [13]. Additionally, the *simple loop* pattern (cf. [14]) is considered. Besides concatenating these patterns, they can be interlaced recursively to create complex workflows respectively complex business processes. An example for such a complex workflow is given in Figure 1. The process steps in this figure are abbreviated with *PS*.

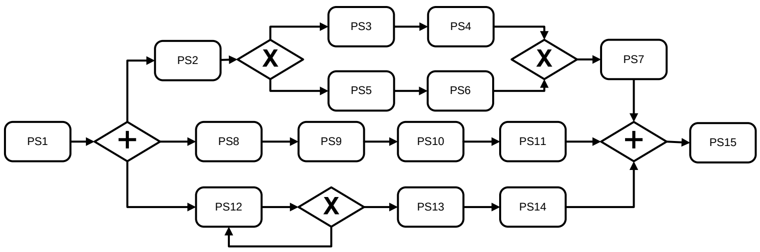

Fig. 1: Example Workflow

The set of all tasks is labeled with I, $i \in I = \{1, ..., n\}$. Each task is accomplished by exactly one service $j \in J_i = \{1, ..., m_i\}$. The decision-variables $x_{ij} \in \{0, 1\}$ state, whether service j is selected for task i or not. As QoS parameters, we take execution time e (the time it takes to execute the service), costs c (costs for the invocation of a service), reliability r (the probability that the service successfully executes), and throughput d (number of parallel service invocations) into account. These parameters – in fact, even a subset of these parameters – are sufficient to cover the aggregation types summation, multiplication and min/max-operator, so that the integration of further QoS parameters into the optimization problem is straightforward. The respective restrictions (bounds) are labeled with $b_{[index]}$. We further consider a pay-per-use pricing model.

When it comes to branchings, we define the set L of paths l as $l \in L = \{1, ..., l^{\#}\}$. Thus, l represents the respective path number within a branching. To make this clear, we refer to Figure 1. After a successful execution of PS_1, the following AND-split branches into three paths l (so $L = \{1, 2, 3\}$), beginning with PS_2, PS_8, PS_{12}. The set $IW_L \subseteq I$ represents the set of tasks within a branching and $IW_l \subseteq IW_L$ the set of tasks within path $l \in L$. The remaining tasks (not located within a branching) are covered in the set $IS = I \backslash (IW_l | l \in L)$. The parameter p_l indicates the probability that path l is executed. We thereby assume $\sum_{l \in L} p_l = 1$.

4 Recursive Workflow Pattern Interlacing

In order to formulate the optimization problem, it is necessary to aggregate the considered QoS parameters and to restrict them to their respective process conditions. Referring to our work in [7], we propose the aggregation functions in Table 1 to address the workflow patterns mentioned in Section 3.

QoS	**Sequence**	**AND-split/-join**	**XOR-split/-join**
e	$\sum_{i \in IS} \sum_{j \in J_i} e_{ij} x_{ij}$	$\max_{l \in L}(\sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij})$	$\sum_{l \in L} p_l \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij}$
c	$\sum_{i \in IS} \sum_{j \in J_i} c_{ij} x_{ij}$	$\sum_{l \in L} \sum_{i \in IW_l} \sum_{j \in J_i} c_{ij} x_{ij}$	$\sum_{l \in L} p_l \sum_{i \in IW_l} \sum_{j \in J_i} c_{ij} x_{ij}$
r	$\prod_{i \in IS} \sum_{j \in J_i} r_{ij} x_{ij}$	$\prod_{l \in L} \prod_{i \in IW_l} \sum_{j \in J_i} r_{ij} x_{ij}$	$\sum_{l \in L} p_l \prod_{i \in IW_l} \sum_{j \in J_i} r_{ij} x_{ij}$
d	$\min_{i \in IS}(\sum_{j \in J_i} d_{ij} x_{ij})$	$\min_{l \in L}(\min_{i \in IW_l} (\sum_{j \in J_i} d_{ij} x_{ij}))$	$\sum_{l \in L} p_l \min_{i \in IW_l} (\sum_{j \in J_i} d_{ij} x_{ij})$

Table 1: Aggregation Functions

But applying these aggregation functions for the optimization implies a sequential arrangement of the process steps within a split and join (cf. [7]). Therefore, we describe in this section an approach to account for recursive interlacings of workflow patterns to overcome this shortcoming. An example for such an interlacing is given in Figure 1. Here, after the AND-split, the process steps in the path starting with PS_2 are not arranged sequentially. Subsequent to PS_2, another split and join follows. To cope with this situation, we abstract from the interlacing by creating a "new" service which represents a composition of the services able to realize the respective tasks within the interlacing. Referring to Figure 1, we build a service by loosely coupling the alternative services for PS_3, PS_4, PS_5, PS_6 according to the structural arrangement of these process steps in the workflow, i.e., XOR-split with XOR-join. This way, we exchange PS_3, PS_4, PS_5, PS_6 for PS_{3456} to obtain a sequential arrangement of the "remaining" process steps PS_2, PS_{3456}, PS_7. To compute the QoS values of the newly composed service, we introduce variables e'_l, c'_l, r'_l, d'_l (with respect to the considered QoS parameters) and apply the appropriate aggregation functions from Table 1 – depending on the kind of split and join. Regarding Figure 1, we specify e'_l, r'_l, d'_l in (1) to (3) – c'_l is specified analogously to e'_l – by applying the respective aggregation functions for XOR-split with XOR-join. Integrating these variables into the optimization problem and performing the described abstraction enables the application of the aggregation functions Table 1 for the optimization.

$$e'_l := \begin{cases} \sum_{l \in L} p_l \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij} & \text{, if interlacing at path } l \\ 0 & \text{, else} \end{cases} \tag{1}$$

$$r'_l := \begin{cases} \sum_{l \in L} p_l \prod_{i \in IW_l} \sum_{j \in J_i} r_{ij} x_{ij} & \text{, if interlacing at path } l \\ 1 & \text{, else} \end{cases} \tag{2}$$

$$d'_l := \begin{cases} \sum_{l \in L} p_l \cdot \min_{i \in IW_l} (\sum_{j \in J_i} d_{ij} x_{ij}) & \text{, if interlacing at path } l \\ \infty & \text{, else} \end{cases} \tag{3}$$

5 Optimization Problem

In this section, we formulate a non-linear optimization problem in Model 1 by specifying the target function in (4) – aiming at minimizing overall costs of the selected services – and restrictions for the aggregated QoS values addressing workflows arbitrarily compiled by combining sequences, AND-splits/-joins, XOR-splits/-joins and Loops. Due to column width, we define $y_{ij} = d_{ij}x_{ij}$ and introduce c_s, c_a, c_x representing the costs for the respective patterns. Further, we use L_a, L_x to seperate AND-splits from XOR-splits. In order to consider Loops, we would exchange the QoS parameters e, c, r for the adapted parameters e^*, c^*, r^* defined in Table 2 (c^* is defined analogously to e^*; d is not affected by a Loop) (cf. [7]).

Model 1 Optimization Problem

Objective Function

$$\text{minimize } F(x) = c_s + c_a + c_x \tag{4}$$

so that

$$\sum_{i \in IS} \sum_{j \in J_i} e_{ij}x_{ij} + \max_{l \in L_a}(e'_l + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij}x_{ij}) + \sum_{l \in L_x} p_l \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij}x_{ij} \leq b_e \tag{5}$$

$$\sum_{i \in IS} \sum_{j \in J_i} c_{ij}x_{ij} + \sum_{l \in L_a}(c'_l + \sum_{i \in IW_l} \sum_{j \in J_i} c_{ij}x_{ij}) + \sum_{l \in L_x} p_l \sum_{i \in IW_l} \sum_{j \in J_i} c_{ij}x_{ij} \leq b_c \tag{6}$$

$$\left(\prod_{i \in IS} \sum_{j \in J_i} r_{ij}x_{ij}\right) \cdot \left(\prod_{l \in L_a} (r'_l \cdot \prod_{i \in IW_l} \sum_{j \in J_i} r_{ij}x_{ij})\right) \cdot \left(\sum_{l \in L_x} p_l \prod_{i \in IW_l} \sum_{j \in J_i} r_{ij}x_{ij}\right) \geq b_r \tag{7}$$

$$\min\left(\min_{i \in IS}(\sum_{j \in J_i} y_{ij}), \min_{l \in L_a}(d'_l, \min_{i \in IW_l}(\sum_{j \in J_i} y_{ij})), \sum_{l \in L_x} p_l \cdot \min_{i \in IW_l}(\sum_{j \in J_i} y_{ij})\right) \geq b_d \tag{8}$$

$$\sum_{l \in L} p_l \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij}x_{ij} = e'_l \qquad \forall l \in L_a | \text{ interlacing at path } l \tag{9}$$

$$\sum_{l \in L} p_l \sum_{i \in IW_l} \sum_{j \in J_i} c_{ij}x_{ij} = c'_l \qquad \forall l \in L_a | \text{ interlacing at path } l \tag{10}$$

$$\sum_{l \in L} p_l \prod_{i \in IW_l} \sum_{j \in J_i} r_{ij}x_{ij} = r'_l \qquad \forall l \in L_a | \text{ interlacing at path } l \tag{11}$$

$$\sum_{l \in L} p_l \cdot \min_{i \in IW_l}(\sum_{j \in J_i} d_{ij}x_{ij}) = d'_l \qquad \forall l \in L_a | \text{ interlacing at path } l \tag{12}$$

$$\sum_{j \in J_i} x_{ij} = 1 \qquad \forall i \in I \tag{13}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in I, \forall j \in J_i \tag{14}$$

In conditions (5) to (8), the restrictions for the regarded QoS parameters are depicted. By integrating e'_l, c'_l, r'_l, d'_l in (5) to (8), we take additional interlacings within the AND-split/-join part into account. The values of e'_l, c'_l, r'_l, d'_l are determined in (9) to (12) by applying the respective aggregation functions for an XOR-split with XOR-join. This way, also other (and more complex) interlacings can be considered. Condition (13) ensures that exactly one service is selected to realize a process step and condition (14) represents the integrality condition.

As the min/max-operator as well as the multiplication are non-linear aggregation types regarding the decision-variables x_{ij}, we require to adapt these non-linear functions and terms in order to obtain a linear optimization problem, which can be solved optimally by applying ILP techniques. To linearize the term with the max-operator in (5), we exchange this term for e_a^{max} and add condition (15) to Model 1. Analogously, additional variables d^{min} are specified and appropriate conditions for each min-operator in (8) are added to Model 1. Regarding condition (7), we apply the approximation in (16) to (7) – which is very accurate for parameter values z_{ij} very close to 1 (such as reliability) – and exchange (7) for (17), which is linear regarding x_{ij}.

$$e_l' + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij} \leq e_a^{max} \qquad \forall l \in L \tag{15}$$

$$\prod_{i \in I} \sum_{j \in J_i} z_{ij} x_{ij} \approx 1 - \sum_{i \in I} (1 - \sum_{j \in J_i} z_{ij} x_{ij}) \tag{16}$$

$$1 - \sum_{l \in L_x} (p_l \sum_{i \in (IS \vee IW_{L_a} \vee IW_l)} (1 - \sum_{j \in J_i} r_{ij} x_{ij})) \geq b_r \tag{17}$$

By performing these adaptation and substitution steps, we transform Model 1 into a linear optimization problem. To compute an optimal solution to this problem – if a solution exists – ILP can be applied. In order to address scalability issues we propose to relax the integrality conditions for the decision-variables x_{ij} and calculate an optimal solution using mixed integer linear programming (MILP) (cf. [6]). A valid (probably non-optimal) solution – containing integer values for x_{ij} – is obtained afterwards by selecting those services, which satisfy the constraints, based on the decision-variables' values. A possible heuristic approach could be H1_RELAX_IP [12], which is not performing significantly worse compared to the optimal solution (cf. [15]).

	Ex. time	**Reliability**
Loop	$e_{ij}^* := \begin{cases} \frac{1}{1-\rho_i} e_{ij} & \text{, if } i \in I_{loop} \\ e_{ij} & \text{, else} \end{cases}$	$r_{ij}^* := \begin{cases} \frac{(1-\rho_i) r_{ij}}{1-\rho_i r_{ij}} & \text{, if } i \in I_{loop} \\ r_{ij} & \text{, else} \end{cases}$

Table 2: Aggregation Functions

6 Conclusion and Outlook

In highly competitive markets, flexible and efficient business process execution is mandatory. By applying service-based processes and selecting services which meet the enterprises' business and QoS requirements best, the mentioned flexibility and efficiency can be increased. This leads to the service-selection-problem which has attracted a lot of research efforts recently. But complex workflows have thereby only insufficiently been addressed in the literature (cf. Section 2). The work at hand enables the computation of the optimal solution to the service-selection-problem without requiring to consider all possible execution paths. It extends previous solutions in this field by allowing arbitrary combinations (including recursive interlacings) of the workflow patterns sequence, AND-split with AND-join, XOR-split with XOR-join, and Loops. We presented an approach to compute an optimal solution to the service-selection-problem by formulating a linear optimization problem for complex workflows, which can be solved optimally using ILP techniques. Applying the optimal service selection increases the mentioned efficiency regarding business process execution.

In our future work, we will focus on integrating security features as qualitative service properties into the optimization. The aim is to arrive at a more or less *secure* communication between service requester, provider and broker, and to ascertain and assess the achieved Quality of Protection (QoP). This achieved QoP will then be considered as additional QoS parameter in the optimization.

Acknowledgements This work is supported in part by E-Finance Lab e. V., Frankfurt am Main, Germany (http://www.efinancelab.com).

References

1. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR, Upper Saddle River, NJ, USA (2004)
2. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. In: Proceedings of WISE. (2003) 3–12
3. Anselmi, J., Ardagna, D., Cremonesi, P.: A QoS-based Selection Approach of Autonomic Grid Services. In: Proceedings of ICSOC. (2007) 1–8
4. Menascé, D.A., Casalicchio, E., Dubey, V.: A Heuristic Approach to optimal Service Selection in Service-oriented Architectures. In: Proceedings of WOSP. (2008) 13–24
5. Huang, A.F.M., Lan, C.W., Yang, S.J.H.: An optimal QoS-based Web Service Selection Scheme. Information Sciences **179**(19) (2009) 3309–3322
6. Domschke, W., Drexl, A.: Einführung in Operations Research. Springer Verlag, Heidelberg (2007)
7. Schuller, D., Eckert, J., Miede, A., Schulte, S., Steinmetz, R.: QoS-Aware Service Composition for Complex Workflows. In: Proceedings of ICIW. (2010)
8. Jaeger, M.C., Rojec-Goldmann, G.: SENECA-Simulation of Algorithms for Selection of Web Services for Composition. In: Proceedings of TES. (2005) 84–97
9. Mabrouk, N.B., Georgantas, N., Issarny, V.: A semantic end-to-end QoS Model for dynamic Service oriented Environments. In: Proceedings of PESOS. (2009) 34–41
10. Ardagna, D., Pernici, B.: Adaptive Service Composition in Flexible Processes. Transactions on Software Engineering **33**(6) (2007) 369–384
11. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-Aware Middleware for Web Services Composition. Transactions on Software Engineering **30**(5) (2004) 311–327
12. Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R.: Heuristics for QoS-aware Web Service Composition. In: Proceedings of ICWS. (2006) 72–82
13. Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed Parallel Databases **14**(1) (2003) 5–51
14. Cardoso, J., Sheth, A.P., Miller, J.A., Arnold, J., Kochut, K.: QoS for Workflows and Web Service Processes. Journal of Web Semantics **1**(3) (2004) 281–308
15. Berbner, R., Spahn, M., Repp, N., Heckmann, O., Steinmetz, R.: Dynamic Replanning of Web Service Workflows. In: Proceedings of DEST. (2007) 211–216