QoS-Aware Service Composition for Complex Workflows

Dieter Schuller, Julian Eckert, André Miede, Stefan Schulte, Ralf Steinmetz
Technische Universität Darmstadt
Multimedia Communications Lab
Rundeturmstr. 10, 64283 Darmstadt, Germany
{schuller, eckert, miede, schulte, steinmetz}@KOM.tu-darmstadt.de

***Abstract*—In Service-oriented Architectures, business processes are realized by composing loosely coupled services. With the ongoing success of Web services technologies, the invocation of Web services goes beyond enterprises' borders. So, besides implementing required services within one enterprise, services can be purchased from external service providers. With a growing number and variety of services, enterprises can choose between functionally equivalent services at different cost and quality levels. The paper at hand addresses this service selection problem considering complex workflow patterns. Aggregation functions for different quality of service parameters are proposed and a linear optimization problem is formulated, which can be solved using integer linear programming techniques or by applying heuristic solutions.**

***Keywords*-Optimization; Service Selection; Quality of Service; Complex Workflows**

I. Introduction

In present times, enterprises generally operate in highly competitive markets, in which a certain number of competing organizations offer similar products and services. In order to sustain themselves in such a market, it is necessary for enterprises to perform their business processes efficiently with respect to costs and quality and to rapidly adapt to changing business requirements within the market. Therefore, a certain degree of flexibility and performance regarding the enterprise's business processes is indispensible. But as nowadays large monolithic software applications, heterogeneous legacy systems, programming languages, operating systems and middleware platforms predominate current enterprise IT architectures, this flexibility can hardly be achieved. In Service-oriented Architectures (SOA), loosely coupled services can be dynamically composed in order to realize business processes and to integrate legacy systems [1]. Each process step of such a business process can be accomplished by a single service. With the emergence of Web service technology, even external partners can be involved into originally internal business processes by inter-connecting Web Services offered from those external partners. This enables an agile and versatile cooperation between several organizations. As a result of dynamically composing services (either provided by external service providers or implemented within the particular organization), business processes can be adapted to the mentioned changing business requirements more quickly by exchanging services appropriately, which increases the flexibility and agility of the enterprise. As different service providers provide services at different Quality of Service (QoS) levels and at different costs, enterprises must decide which services from which service providers to select in order to realize its business processes. In other words, an organization has the opportunity to select those Web Services that meet its business and QoS requirements best.

The problem of selecting appropriate services that meet specified conditions and restrictions on business process QoS and costs (Service Selection Problem) has been discussed by several authors recently (cf. [2], [3]). With respect to workflow patterns, it has to be noted that sequential workflows only cover a subset of existing workflows in reality [4]. This is what we have seen in various industry projects. Therefore, we address the optimization of service compositions in complex workflows.

The remainder of the paper at hand is organized as follows. In Section II, we present related work. The system model, where we make basic assumptions and specify the considered parameters, is described in Section III. Here, we also introduce the applied cost model. Our optimization approach is discussed in Section IV and a linear optimization problem will be formulated in Section V. Finally, in Section VI, we draw conclusions and discuss future work.

II. Related Work

A lot of work regarding different workflow patterns and workflow modeling has been done by van der Aalst and van Hee in [5]. In their work [4], Cardoso et al. propose aggregation functions for some of those workflow patterns. Also, the work of Jaeger et al. ([6], [7]) addresses the aggregation of QoS values for Web service compositions. The authors in [8], [9] specify an optimization problem to solve the QoS-aware service selection problem for multiple execution paths by merging the optimal execution plans for each possible execution path. They also provide an integer linear programming (ILP) solution, but they do not consider probabilities for possible execution paths. A heuristic solution to the service selection problem is proposed by the authors in [10], [11]. A replanning mechanism, which re-estimates the workflow QoS if further information as, e.g., the decision for one of the possible executions paths

is available or if SLA violations become more likely, is introduced in [12]. In their work [13], Berbner et al. perform a replanning in addition to SLA violations, if Web Services perform better than committed in its SLAs to reduce costs.

In the paper at hand, we present an approach for formulating an ILP problem for the service selection problem – which can be solved using ILP techniques (cf. [14]) – considering necessary probabilities in complex workflows. This extends the work by [8], [9], as our approach provides a basis for multiple combinations of complex workflow patterns and does not require the knowledge of all possible execution paths in advance. In addition, the heuristic solution method proposed in [15] can be applied to our approach in order to address scalability issues.

III. System Model

In this section, we describe the system model that is necessary to formulate a linear optimization problem, which can be solved using ILP techniques. Therefore, we specify an objective function in Section V and discuss the necessary process conditions in Section IV. Due to complexity issues, we concentrate in the paper at hand on the workflow patterns sequence, parallel split (AND-split), exclusive choice (XOR-split), and structured loop, which only form a subset of possible workflows. We assume an abstract execution plan (e.g., written in BPEL) consisting of n abstract process steps respectively tasks (we will use these terms synonymously). For each process step PS_i with $i \in I = \{1, ..., n\}$, a set J_i of m_i alternative Web services $j \in J_i = \{1, ..., m_i\}$ exist, which are assumed to be able to provide the functionally needed by task PS_i. Further, each task PS_i is accomplished by exactly one service $j \in J_i$. The decision variables $x_{ij} \in \{0, 1\}$ state, whether Web service j is selected for task PS_i. As already mentioned in the previous section, besides costs for the invocation of one service, we only consider execution time (the time it takes to execute the service), reliability (the probability that the service successfully provides the requested results), and throughput (the number of parallel service invocations), as these parameters are sufficient to cover the aggregation types summation, multiplication and the min/max operator (the integration of further QoS parameters is straightforward). Thereby, we consider a pay-per-use pricing model. With respect to the exclusive choice and the loop pattern, we need to take the probabilities for the execution of a certain path into account. For this, we perform an average-case, best-case and worst-case analysis for the selected workflow patterns. From the abstract execution plan, we depict the chronological order of the tasks as follows: if PS_k is a direct successor of PS_i, we add $PS_i \rightarrow PS_k$ to a set $DS = \{PS_i \rightarrow PS_k | PS_k \text{ direct successor of } PS_i\}$. DS_s is the set of *start* tasks, i.e., the tasks that need to be executed first in the execution plan. In addition, we define DS_e as the set of *end* tasks, i.e., tasks with no direct successor.

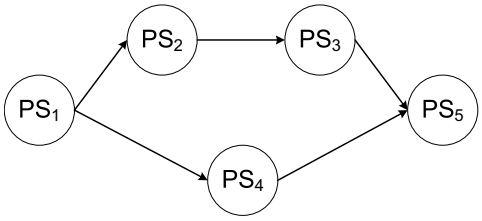

Figure 1. Split-Join

IV. Aggregation Functions

In this section, we describe our approach by specifying the aggregation functions (with respect to the mentioned QoS parameters) which are necessary for the conditions of the optimization problem in Section V.

A. Execution Time

With respect to execution time, we define t_i as the time in which task PS_i begins its execution. The execution time of the service j_i that executes task PS_i is denoted as e_{ij}. Let r_e denote the process restriction for execution time.

1) Sequence and Parallel Split: In the following, we analyze the aggregation of execution times for the workflow patterns *sequence* and *parallel split*. Therefore, we add conditions (1), (2), and (3) to the optimization problem, which is discussed in Section V.

$$t_i = 0 \qquad \forall i \in I | PS_i \in DS_s \tag{1}$$

$$t_i + \sum_{j=1}^{m_i} e_{ij} x_{ij} \leq t_k \qquad \forall i \in I | PS_i \rightarrow PS_k \in DS \tag{2}$$

$$t_i + \sum_{j=1}^{m_i} e_{ij} x_{ij} \leq r_e \qquad \forall i \in I | PS_i \in DS_e \tag{3}$$

In equation (1) we specify that the execution of the first tasks start at time 0. This condition is necessary to compare time t_i with the restriction for the execution time r_e in equation (3). Equation (2) makes sure that task PS_k cannot be started before all precedent tasks PS_i have been started and finished their executions, which is expressed by the sum of the starting period t_i of task PS_i and its execution time. Finally, equation (3) demands compliance with the process restriction r_e for the execution time.

2) Exclusive Choice: With respect to the workflow pattern *exclusive choice*, we need to consider the amount of possible paths resulting from the XOR-split. E.g., in our example workflow in Figure 1, we need to consider the two possible paths $PS_1 \rightarrow PS_2 \rightarrow PS_3 \rightarrow PS_5$ and $PS_1 \rightarrow PS_4 \rightarrow PS_5$.

In this respect we consider a total amount of o paths with $l \in L = \{1, ..., o\}$. So, l represents the respective path number. As things get more complicated when it comes to combinations of XOR-splits, we will use l to express only the number of the XOR-part of the paths and

name it XOR-path. For each possible XOR-path, there is a set of tasks $PS_{i_l} \in W_l = \{PS_i | PS_i \text{ in XOR-path } l\}$ representing the tasks of the path which is exclusively executed. The respective task numbers are labeled with $i_l \in IW_l = \{i | PS_i \in W_l\}$. We assume a sequential workflow pattern (other cases are discussed in Section VI) for the tasks $PS_{i_l} \in W_l$. Thereby, we name the first task in this sequence $PS^1_{i_l}$ and the last one $PS^e_{i_l}$. To make this clear, we refer to Figure 1. As already mentioned, there are two possible XOR-paths ($l = 1$ and $l = 2$) with $W_1 = \{PS_2, PS_3\}$ and $W_2 = \{PS_4\}$, $PS^1_{i_1} = PS_2$ and $PS^e_{i_1} = PS_3$, and $PS^1_{i_2} = PS^e_{i_2} = PS_4$.

The probability for executing XOR-path l is indicated by p_l with $\sum_{l=1}^{o} p_l = 1$. We assume the last process steps of each path $PS^e_{i_l}$ precede a certain process step PS_k (we otherwise would insert an overall dummy end-task). In other words, we assume XOR-join before PS_k, which is depicted in (4). For the *average-case analysis*, we propose (5).

$$PS^e_{i_l} \rightarrow PS_k \qquad \forall PS^e_{i_l} \in W_l \tag{4}$$

$$\sum_{l \in L} p_l (t^1_{i_l} + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij}) \leq t_k \tag{5}$$

In equation (5), we calculate the sum of the execution times of each possible XOR-path l and add the start time of the first process steps $t^1_{i_l}$, respectively. These results are weighted with the respective probabilities p_l and summed up to achieve the average starting time of process step PS_k. If there is no XOR-join in the abstract execution plan (as assumed in (4)), we exchange (5) for (6).

$$\sum_{l \in L} p_l (t^1_{i_l} + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij}) \leq r_e \tag{6}$$

As we here consider an XOR-split without an XOR-join, there will be no join at all (except for further splits). In other words, there are no direct successors for $PS^e_{i_l}$. So, $PS^e_{i_l} \in DS_e$. Therefore, the calculated weighted sum in (6) has to be lower than or equal to the allowed restriction r_e for execution time.

With respect to a *best-case analysis*, we select the XOR-path with the lowest execution time. Therefore, we replace (5) with (7) and (6) with (8).

$$\min_{l \in L} \{(t^1_{i_l} + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij})\} \leq t_k \tag{7}$$

$$\min_{l \in L} \{(t^1_{i_l} + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij})\} \leq r_e \tag{8}$$

Equations (7) and (8) indicate that only the shortest XOR-path in terms of execution time needs to fulfill the restrictions. With respect to a *worst-case analysis*, we select the XOR-path with the highest execution time. Therefore, we replace (5) by (9) and (6) by (10).

$$\max_{l \in L} \{(t^1_{i_l} + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij})\} \geq t_k \tag{9}$$

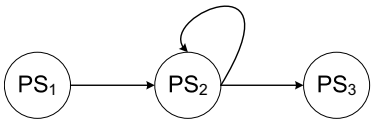

Figure 2. Structured Loop

$$\max_{l \in L} \{(t^1_{i_l} + \sum_{i \in IW_l} \sum_{j \in J_i} e_{ij} x_{ij})\} \geq r_e \tag{10}$$

3) Structured Loop: With respect to the workflow pattern *structured loop* we again have to consider probabilities. In order to distinguish the probabilities in the context of loop from the probabilities of the exclusive choice, we use ρ for the former ones. Referring to Figure 2, ρ is the probability that PS_2 is executed another time.

After a second execution, it is again possible to do the execution another time with probability ρ. The aggregated probability for this is $\rho \cdot \rho = \rho^2$. So, in general, with regard to a structured loop at PS_i, the probability that the service realizing PS_i is executed n times is ρ^{n-1}. If the total number of executions ae of the respective service was known before the optimal execution plan is calculated, we would add equation (11) to the set of process conditions. Thereby, we define $LOOP$ as the set of process steps PS_i, I_{loop} as the set of process step numbers i, where a loop is located, and ρ_i as the respective probability. In addition, we assume $PS_i \rightarrow PS_k$ with $PS_i \in LOOP$.

$$t_i + ae \sum_{j \in J_i} e_{ij} x_{ij} \leq t_k \qquad \forall i \in I_{loop} \tag{11}$$

In case the total number of executions is not known beforehand, so it (the amount of executions) only depends on ρ, we change (11) to (12). By applying the formula for geometric series, we come to an *average-case analysis* in equation (13).

$$t_i + \sum_{f=1}^{\infty} \rho_i^{f-1} \sum_{j \in J_i} e_{ij} x_{ij} \leq t_k \qquad \forall i \in I_{loop} \tag{12}$$

$$t_i + \frac{1}{1 - \rho_i} \sum_{j=1}^{m_i} e_{ij} x_{ij} \leq t_k \qquad \forall i \in I_{loop} \tag{13}$$

Performing a *best-case analysis*, process steps $PS_i \in LOOP$ are only executed once. So we simply set ae to 1, which changes equation (11) to (2). In other words, we omit the loop pattern. In a *worst-case analysis*, the respective services are executed endlessly, which can be expressed by setting ae to ∞. In this case, the process execution will not terminate at all. That is why it does not make sense to perform a worst-case analysis with loops here.

B. Costs

With respect to costs, we define c_{ij} as the cost resulting from executing service $j \in J_i$ to accomplish process step PS_i. The process restriction for costs is denoted by r_c.

1) Sequence and Parallel Split: Regarding the aggregation of cost, we add condition (14) to the optimization problem (cf. Section V) for the patterns *sequence* and *parallel split*.

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}x_{ij} \leq r_c \tag{14}$$

As for each process step a service is invoked, the costs for each of the invoked services have to be summed up.

2) Exclusive Choice: In order to provide process restrictions for the workflow pattern *exclusive choice*, we have to distinguish the process steps within the XOR-split ($PS_i \in W_l$) from the other ones, which are assumed to be arranged sequentially. So we build a set S of process steps PS_i by removing process steps $PS_i \in W_l$ from the set of *all* process steps. So, $S = \{PS_1, ..., PS_n\} \backslash (W_1 \vee ... \vee W_o)$. For the set of process step numbers IS, we do the same. So, $IS = I \backslash (IW_1 \vee ... \vee IW_o)$. Referring to the former parameter definitions , we propose conditions (15), (16), and (17).

$$\sum_{i\in IS}\sum_{j=1}^{m_i} c_{ij}x_{ij} + \sum_{l\in L} p_l \sum_{i\in IW_l}\sum_{j\in J_i} c_{ij}x_{ij} \leq r_c \tag{15}$$

$$\sum_{i\in IS}\sum_{j=1}^{m_i} c_{ij}x_{ij} + \min_{l\in L}\{\sum_{i\in IW_l}\sum_{j\in J_i} c_{ij}x_{ij}\} \leq r_c \tag{16}$$

$$\sum_{i\in IS}\sum_{j=1}^{m_i} c_{ij}x_{ij} + \max_{l\in L}\{\sum_{i\in IW_l}\sum_{j\in J_i} c_{ij}x_{ij}\} \leq r_c \tag{17}$$

In the *average-case analysis*, which is depicted in (15), we consider the weighted sum of costs for the different XOR-paths. In order to perform a *best-case analysis*, we only consider the XOR-path in (16), where the aggregated costs for service invocation are lowest. The opposite is true for the *worst-case analysis* in (17).

3) Structured Loop: If the amount of executions ae is known beforehand, we define c^h_{ij} in (18) and propose condition (19).

$$c^h_{ij} := \begin{cases} ae \cdot c_{ij} & \text{, if } i \in I_{loop} \\ c_{ij} & \text{, else} \end{cases} \tag{18}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} c^h_{ij}x_{ij} \leq r_c \tag{19}$$

In case ae is unknown we define c^*_{ij} in (20) by applying the formula for geometric series in analogy to (12) and (13) and propose condition (21) for *average-case analysis*.

$$c^*_{ij} := \begin{cases} \frac{1}{1-\rho_i} c_{ij} & \text{, if } i \in I_{loop} \\ c_{ij} & \text{, else} \end{cases} \tag{20}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} c^*_{ij}x_{ij} \leq r_c \tag{21}$$

Again, a best-case analysis can be performed by executing the respective services only once, which is equal to setting $ae = 1$ in (18) and adding (19) to the set of conditions of the optimization problem. As already mentioned, in a *worst-case analysis*, the execution of the respective services will not stop. For this, a worst-case analysis in not applied here.

C. Reliability

With respect to reliability, we define r_{ij} as the reliability of service $j \in J_i$ accomplishing process step PS_i. The process restriction for reliability is denoted by r_r.

1) Sequence and Parallel Split: Similar to costs (cf. (14)), the aggregation of reliability is depicted in (22) for the patterns *sequence* and *parallel split* and added to the optimization problem in Section V.

$$\prod_{i=1}^{n}\sum_{j=1}^{m_i} r_{ij}x_{ij} \geq r_r \tag{22}$$

2) Exclusive Choice: For an *average-case analysis*, we propose condition (23). In conditions (24) and (25), a *best-case* and *worst-case* analysis is depicted, as we select the XOR-path l with the highest/lowest aggregated reliability.

$$(\prod_{i\in IS}\sum_{j\in J_i} r_{ij}x_{ij}) \cdot (\sum_{l\in L} p_l(\prod_{i\in IW_l}\sum_{j\in J_i} r_{ij}x_{ij})) \geq r_r \tag{23}$$

$$(\prod_{i\in IS}\sum_{j\in J_i} r_{ij}x_{ij}) \cdot \max_{l\in L}\{(\prod_{i\in IW_l}\sum_{j\in J_i} r_{ij}x_{ij})\} \geq r_r \tag{24}$$

$$(\prod_{i\in IS}\sum_{j\in J_i} r_{ij}x_{ij}) \cdot \min_{l\in L}\{(\prod_{i\in IW_l}\sum_{j\in J_i} r_{ij}x_{ij})\} \geq r_r \tag{25}$$

3) Structured Loop: In analogy to (18), we define r^h_{ij} in (26) and propose condition (27).

$$r^h_{ij} := \begin{cases} (r_{ij})^{ae} & \text{, if } i \in I_{loop} \\ r_{ij} & \text{, else} \end{cases} \tag{26}$$

$$\prod_{i=1}^{n}\sum_{j=1}^{m_i} r^h_{ij}x_{ij} \leq r_r \tag{27}$$

In case ae is unknown we similarly define r^*_{ij} in (28) and provide condition (29) for an *average-case analysis*. For a *best-case analysis*, we set $ae = 1$ in (26) and add (27) to the set of conditions of the optimization problem. Performing a worst-case analysis is impossible.

$$r^*_{ij} := \begin{cases} \frac{(1-\rho_i)r_{ij}}{1-\rho_i r_{ij}} & \text{, if } i \in I_{loop} \\ r_{ij} & \text{, else} \end{cases} \tag{28}$$

$$\prod_{i=1}^{n}\sum_{j=1}^{m_i} r^*_{ij}x_{ij} \leq r_r \tag{29}$$

To explain (28), we need to consider all potential scenarios. The loop can be performed once, twice, and so on – or not at all. So the resulting reliability $r^*_{ij} = (1-\rho_i)r_{ij} + (1-\rho_i)\rho_i(r_{ij})^2 + (1-\rho_i)(\rho_i)^2(r_{ij})^3 + ... = \frac{\rho_i}{\rho_i}(1-\rho_i)r_{ij} + \frac{\rho_i}{\rho_i}(1-\rho_i)\rho_i(r_{ij})^2 + \frac{\rho_i}{\rho_i}(1-\rho_i)(\rho_i)^2(r_{ij})^3 + ... = \frac{1-\rho_i}{\rho_i}((\rho_i r_{ij})^1 + (\rho_i r_{ij})^2 + (\rho_i r_{ij})^3 + ...) = \frac{1-\rho_i}{\rho_i}\rho_i r_{ij}(1 + (\rho_i r_{ij})^1 + (\rho_i r_{ij})^2 + ...) = (1-\rho_i)r_{ij}\frac{1}{1-\rho_i r_{ij}}$.

D. Throughput

Regarding throughput, d_{ij} is the throughput of service $j \in J_i$ realizing process step PS_i. The restriction for throughput is labeled with r_d.

1) Sequence and Parallel Split: In the workflow patterns sequence and parallel split, the throughput of each service realizing one of the tasks PS_i has to exceed the process restriction r_d, which is depicted in condition (30).

$$\min_{i=1}^{n}\{\sum_{j=1}^{m_i} d_{ij}x_{ij}\} \geq r_d \tag{30}$$

2) Exclusive Choice: For an average-case analysis, we propose (31). Conditions (32) and (33) refer to a best-case and worst-case analysis, as we select the maximum, respectively, the minimum possible throughput. In order not to exceed the column width, we define $y_ij = d_{ij}x_{ij}$.

$$\min\{\min_{i\in IS}\{\sum_{j\in J_i} y_{ij}\}, \sum_{l\in L} p_l \min_{i\in IW_l}\{\sum_{j\in J_i} y_{ij}\}\} \geq r_d \tag{31}$$

$$\min\{\min_{i\in IS}\{\sum_{j\in J_i} y_{ij}\}, \max_{l\in L}\{\min_{i\in IW_l}\{\sum_{j\in J_i} y_{ij}\}\}\} \geq r_d \tag{32}$$

$$\min\{\min_{i\in IS}\{\sum_{j\in J_i} y_{ij}\}, \min_{l\in L}\{\min_{i\in IW_l}\{\sum_{j\in J_i} y_{ij}\}\}\} \geq r_d \tag{33}$$

3) Structured Loop: As the amount of executions of a certain service does not affect the throughput of the workflow, condition (30) also covers the workflow pattern *structured loop*.

V. Optimization Problem

For the formulation of a linear optimization problem, we conduct an average-case analysis in the paper at hand. Further, we consider the workflow patterns sequence, parallel split, exclusive choice and structured loop. With respect to the objective function, it has to be noted that in a multi-dimensional objective function (as we proposed in [16]), only normalized QoS values of those services, which are on the *critical* path for the regarded QoS parameter, have to be considered. Otherwise a service's execution time is for instance taken into account, which probably does not matter, regarding the objective function. To explain this, we assume a scenario, where the execution time of a service does not matter (as this service is not on a critical path (cf., e.g., [8])). In such a case, the service selection decision would be negatively influenced, as the costs of this service and its execution time (which does not matter as assumed) are reckoned up, although the execution time is irrelevant in this case and should have been ignored therefore. This is why we consider a single-dimensional objective function in (34), which minimizes the total overall costs for the workflow execution, and propose Model 1. Due to the restricted column-width, we define a set IDS_l^e for process step numbers $i_l \in IDS_l^e = \{i|PS_{i_l}^e \rightarrow PS_k \in DS\}$.

Model 1 Optimization Problem

Objective Function

$$\text{minimize } F(x) = \sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}^* x_{ij} \tag{34}$$

s.t.

$$t_i = 0 \qquad \forall i \in I|PS_i \in DS_s \tag{35}$$

$$t_i + \sum_{j=1}^{m_i} e_{ij}x_{ij} \leq t_k \qquad \forall i \in I|PS_i \rightarrow PS_k \in DS \tag{36}$$

$$t_i + \sum_{j=1}^{m_i} e_{ij}x_{ij} \leq r_e \qquad \forall i \in I|PS_i \in DS_e \tag{37}$$

$$\sum_{l=1}^{o} p_l(t_{i_l}^1 + \sum_{i\in IW_l}\sum_{j\in J_i} e_{ij}x_{ij}) \leq t_k \qquad \forall i_l \in IDS_l^e \tag{38}$$

$$\sum_{l=1}^{o} p_l(t_{i_l}^1 + \sum_{i\in IW_l}\sum_{j\in J_i} e_{ij}x_{ij}) \leq r_e \forall i \in I|PS_{i_l}^e \in W_l \tag{39}$$

$$t_i + \frac{1}{1-\rho_i}\sum_{j=1}^{m_i} e_{ij}x_{ij} \leq t_k \qquad \forall i \in I_{loop} \tag{40}$$

$$\sum_{i\in IS}\sum_{j\in J_i} c_{ij}^* x_{ij} + \sum_{l\in L} p_l \sum_{i\in IW_l}\sum_{j\in J_i} c_{ij}^* x_{ij} \leq r_c \tag{41}$$

$$(\prod_{i\in IS}\sum_{j\in J_i} r_{ij}^* x_{ij}) \cdot (\sum_{l\in L} p_l(\prod_{i\in IW_l}\sum_{j\in J_i} r_{ij}^* x_{ij})) \geq r_r \tag{42}$$

$$\min\{\min_{i\in IS}\{\sum_{j\in J_i} y_{ij}\}, \sum_{l\in L} p_l \min_{i\in IW_l}\{\sum_{j\in J_i} y_{ij}\}\} \geq r_d \tag{43}$$

$$\sum_{j=1}^{m_i} x_{ij} = 1 \qquad \forall i \in I \tag{44}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in I, \forall j \in J_i \tag{45}$$

In order to develop a *linear* optimization problem, we need to exchange the non-linear terms in (42) and (43). Therefore, we use the approximation in (46), which is very accurate for parameter values z_{ij} close to 1, to exchange (42) by (47).

$$\prod_{i=1}^{n}\sum_{j=1}^{m_i} z_{ij}x_{ij} \approx 1 - \sum_{i=1}^{n}(1 - \sum_{j=1}^{m_i} z_{ij}x_{ij}) \tag{46}$$

$$1 - \sum_{l\in L}(p_l \sum_{i\in(IS\vee IW_l)}(1 - \sum_{j\in J_i} r_{ij}^* x_{ij})) \geq r_r \tag{47}$$

With respect to throughput, we exchange (43) by (48) and (49). To explain this, it has to be noted that if the minimum of a set has to be greater or equal to a certain threshold, each element of this set has to fulfill the constraint.

$$\sum_{j \in J_i} d_{ij} x_{ij} \geq r_d \qquad \forall i \in IS \tag{48}$$

$$\sum_{l \in L} p_l \min_{i \in IW_l} \{ \sum_{j \in J_i} d_{ij} x_{ij} \} \geq r_d \tag{49}$$

As there are still non-linear terms in (49), we would have to compute the power set for each possible combination of process steps from the different XOR-paths l and restrict the weighted sum for each combination to be greater or equal to r_d. So, we need to add a condition like (50) for each combination $(PS_{i_{l_1}}, PS_{i_{l_2}})$ to replace the *min* operator.

$$\sum_{l \in L} p_l \{ \sum_{j \in J_{i_{l_1}}} y_{ij}, \sum_{j \in J_{i_{l_2}}} y_{ij} \} \geq r_d \quad \forall (PS_{i_{l_1}}, PS_{i_{l_2}}) \tag{50}$$

As the amount of possible combinations grows exponentially with the number of process steps for each XOR-path, we propose (as an alternative to adding this amount of restrictions) to increase the restriction strength for r_d by exchanging (43) by (51) instead of adding the mentioned conditions (50).

$$\sum_{j \in J_i} d_{ij} x_{ij} \geq r_d \qquad \forall i \in I \tag{51}$$

After exchanging (42) and (43) by (47), (49) and (50), or (47) and (51), the presented optimization problem becomes a linear optimization problem, which can be solved using ILP techniques (cf. [14]). As it requires strong computational effort to calculate the optimal solution with a growing number of process steps and alternative services per process step, we propose to relax the integrality condition (45) and calculate an optimal solution using mixed integer linear programming (MLIP) techniques. Afterwards, we obtain a valid, probably non-optimal solution containing (solely) integer values by selecting services based on its decision varablesvalues, which satisfy constraints. A possible heuristic approach could be H1_RELAX_IP [15], which is not performing significantly worse compared with the optimal solution (cf. [13]).

VI. Conclusion and Future Work

In the paper at hand, we formulated a linear optimization problem solving the service selection problem for complex workflows. As we assumed sequential workflow execution within split and joined paths, we focus on interlacing of those workflow patterns in our future work. So far, we discovered that recursively applying the proposed aggregation functions leads to a feasible solution to this problem. In addition, we will formulate the optimization problem with respect to a best-case and worst-case analysis.

Acknowledgment

This work is supported in part by the BMBF-sponsored project SoKNOS (http://www.soknos.de) and the E-Finance Lab e. V., Frankfurt am Main, Germany. (http://www.efinancelab.com).

References

[1] M. P. Papazoglou, "Service-Oriented Computing: Concepts, Characteristics and Directions," in *Web Information Systems Engineering*, 2003, pp. 3–12.

[2] T. Yu and K.-J. Lin, "Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints," in *Intl. Conference on Service-oriented Computing*, 2005, pp. 130–143.

[3] D. Ardagna, G. Giunta, N. Ingraffia, R. Mirandola, and B. Pernici, "QoS-Driven Web Services Selection in Autonomic Grid Environments," in *OTM Conferences (2)*, 2006, pp. 1273–1289.

[4] J. Cardoso, A. P. Sheth, J. A. Miller, J. Arnold, and K. Kochut, "QoS for Workflows and Web Service Processes." *Journal of Web Semantics*, vol. 1, no. 3, pp. 281–308, 2004.

[5] W. M. van der Aalst and K. M. van Hee, *Workflow Management: Models, Methods, and Systems.* MIT Press, 2002.

[6] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl, "QoS Aggregation for Web Service Composition using Workflow Patterns," in *Enterprise Distributed Object Computing Conference*, 2004, pp. 149–159.

[7] M. C. Jaeger, G. Mühl, and S. Golze, "QoS-Aware Composition of Web Services: A Look at Selection Algorithms," in *Intl. Conference on Web Services*, 2005, pp. 807–808.

[8] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-Aware Middleware for Web Services Composition." *Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[9] D. Ardagna and B. Pernici, "Adaptive Service Composition in Flexible Processes," *IEEE Trans. Software Eng.*, vol. 33, no. 6, pp. 369–384, 2007.

[10] J. Anselmi, D. Ardagna, and P. Cremonesi, "A QoS-based Selection Approach of Autonomic Grid Services," in *Intl. Conference on Service-oriented Computing*, 2007, pp. 1–8.

[11] D. A. Menascé, E. Casalicchio, and V. Dubey, "A Heuristic Approach to optimal Service Selection in Service-oriented Architectures," in *Workshop on Software and Performance*, 2008, pp. 13–24.

[12] G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, "QoS-Aware Replanning of Composite Web Services," in *Intl. Conference on Web Services*, 2005, pp. 121–129.

[13] R. Berbner, M. Spahn, N. Repp, O. Heckmann, and R. Steinmetz, "Dynamic Replanning of Web Service Workflows," in *Digital Ecosystems and Technologies*, 2007.

[14] W. Domschke and A. Drexl, *Einführung in Operations Research.*, 7th ed. Springer Verlag, Heidelberg, 2007.

[15] R. Berbner, M. Spahn, N. Repp, O. Heckmann, and R. Steinmetz, "Heuristics for QoS-aware Web Service Composition," in *Intl. Conference on Web Services*, 2006.

[16] D. Schuller, A. Papageorgiou, S. Schulte, J. Eckert, N. Repp, and R. Steinmetz, "Process Reliability in Service-oriented Architectures," in *Digital Ecosystems and Technologies*, 2009.