1-1-2009

Dynamic service selection for service composition with time constraints

Azlan Ismail
University of Wollongong, ai423@uow.edu.au

Jun Yan
University of Wollongong, jyan@uow.edu.au

Jun Shen
University of Wollongong, jshen@uow.edu.au

Recommended Citation

Ismail, Azlan; Yan, Jun; and Shen, Jun: Dynamic service selection for service composition with time constraints 2009, 183-190.
https://ro.uow.edu.au/infopapers/1702

Dynamic service selection for service composition with time constraints

Abstract

Service selection is a kind of planning approach that evaluates and selects from multiple services to form a composite plan. In service selection, we found one additional issue that has not been investigated yet namely time constraints consistency among composite services. This issue is significant because there are potential time constraints involved that might cause inconsistency between selected services although they can offer aggregated QoS that satisfy global requirements. Furthermore, there might be some unintended waiting time to be considered in the QoS aggregation. Thus, this paper contributes to the analysis on the problem caused by the time constraints and proposes general selection approach to tackle these issues. The approach comprises two major functionalities; (i) exploration of the process model based on patterns, (ii) evaluation of candidates based on time constraints and objective functions.

Keywords

era2015, Dynamic, service, selection, for, service, composition, time, constraints

Disciplines

Physical Sciences and Mathematics

Publication Details

Ismail, A., Yan, J. & Shen, J. (2009). Dynamic service selection for service composition with time constraints. Australian Software Engineering Conference (pp. 183-190). Los Alamitos, USA: IEEE.

Dynamic Service Selection for Service Composition with Time Constraints

Azlan Ismail, Jun Yan, Jun Shen
School of Information Systems and Technology
University of Wollongong
NSW, 2522, Australia
{*ai423,jyan,jshen*}*@uow.edu.au*

Abstract

Service selection is a kind of planning approach that evaluates and selects from multiple services to form a composite plan. In service selection, we found one additional issue that has not been investigated yet namely time constraints consistency among composite services. This issue is significant because there are potential time constraints involved that might cause inconsistency between selected services although they can offer aggregated QoS that satisfy global requirements. Furthermore, there might be some unintended waiting time to be considered in the QoS aggregation. Thus, this paper contributes to the analysis on the problem caused by the time constraints and proposes general selection approach to tackle these issues. The approach comprises two major functionalities; (i)exploration of the process model based on patterns, (ii)evaluation of candidates based on time constraints and objective functions.

1. Introduction

Web services are self-contained modules - deployed over standard middleware platforms - that can be described, published, located, orchestrated, and programmed using XML-based technologies over a network [1]. One of the most distinct advantages of the web services technology is that existing services can be composed into a loosely coupled network of services, i.e., a business process, to form a value added composite service. Each activity in such process is associated with a service from a provider to perform the expected business operations. This capability can reduce the time required to develop and implement a business process.

Generally, services are selected from a large set of services which involve multiple service providers. Each service has its own descriptions for advertising its capabilities. The basis of service selection is to choose services that are able to provide operation for the entire process based on the advertised information. Obviously, selected services can have considerable impact on overall process execution quality, as different providers may offer the same function with different quality levels. Hence, the critical selection criterion is about the non-functional properties of services, namely Quality of Service (QoS) attributes (e.g., time, price).

Existing works on the service selection have concentrated on QoS aggregation to support service selection. The ultimate goal is to satisfy user requirements especially those that are imposed at the composite or application level. Despite these efforts, we have an innovative issue where there are potential time constraints introduced by individual services. These time constraints have to be checked for the consistency among services to be selected. Furthermore, there might be some unintended waiting times to be considered in the aggregation. This additional issue makes the existing proposal of service selection insufficient as we illustrated in the motivation. Therefore an innovative approach is very much needed.

We propose a service selection approach that tackles this issue with the aim at finding the most feasible selection plan. The approach comprises two major functionalities; (i) Exploration of process model based on patterns (ii) Evaluation of candidates based on time constraints and objective functions.

The first function is essential to guide the construction of the selection plan. The second function is meant to find consistent combination which can satisfy global requirements. Consistency is determined based on the scheduling generation. Meanwhile, the global requirement satisfaction is determined based on two major procedures; ranking and aggregation. In general, ranking procedure determines the best value or candidate for each set. Then the selected candidates are aggregated and checked with the objective value. If the aggregated value satisfies the objective value, then a feasible selection plan is found. Otherwise, a backtracking method will be executed which revert the plan to the previous decision. However, backtracking is beyond the scope of this paper.

The remainder of this paper is organized as follows. Section 2 illustrates the motivation of this work. Section 3 presents our selection approach. Section 4 discusses the approach in a case study. Section 5 remarks existing works. Section 6 concludes this paper and outlines our future works.

2. Motivation

To illustrate the motivation of this research, we use an example of supply chain process as depicted in Fig. 1.

1530-0803/09 $25.00 © 2009 IEEE
DOI 10.1109/ASWEC.2009.30

IEEE computer society

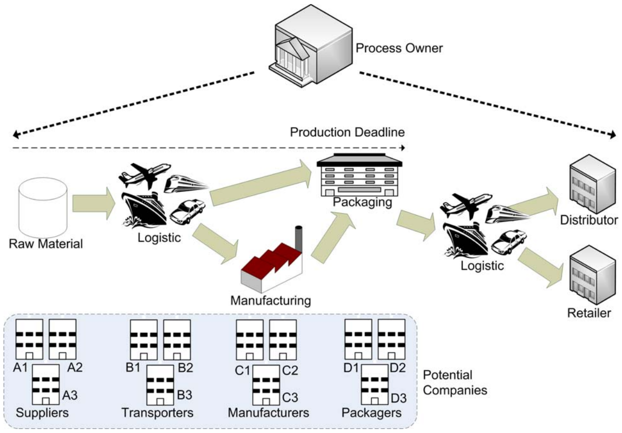


Figure 1. A supply chain scenario for specific business project which involves external companies to execute the business activities.

The process represents specific business project owned by a big company which also acts as the project owner. The business project comprises of typical supply chain activities namely, getting materials, delivering materials, manufacturing, packaging and distributing the products. Each activity will be executed by potential external companies identified by the project owner with certain criteria. In relation to the service selection, the project owner is responsible for selecting external services that are suitable to participate in the process.

In this scenario, assume we are interested in selecting companies that can collaboratively meet the production deadline. Therefore, the selection involves deciding which suppliers, transporters, manufactures and packagers to be selected in the business project. It is the responsibility of the project owner to find the suitable selection that can collaboratively meets the deadline. The basis of the selection is by considering the QoS offered by these companies. In general, a selection plan can be obtained by aggregating the QoS values. A plan that satisfies the deadline will be selected.

However, there is one additional issue that has to be considered in the selection plan. This issue is the availability of the potential companies to participate in specific business project. In reality, external companies are constrained by commitment with other business projects. For instance, transporter B2 can offer fast delivery but only can participate on 1st September due to commitment with other business project. Another example, manufacturer C3 can offer fast production but has to finish by 5th September to cater other production of different business project. This situation can be seen as time constraint which constrained the participation of the companies.

As individual companies, the time constraint will not cause any problem. However, if both are selected, then the time constraints may cause severe problems. To understand the problem, assume the process owner is at the stage of deciding which transporters (B group) and manufacturers (C group) are suitable to be selected in the business project. A sample of data related to QoS offered and time constraints are given in Fig. 2.

Comp. ID	Processing-Time	Start-After/On	Finish-Before/By
B1	3 day	1st Sep 2008	depend on processing-time
B2	2 day	3rd Sep 2008	depend on processing-time
B3	4 day	31st Aug 2008	6th Sep 2008
C1	6 day	7th Sep 2008	13th Sep 2008
C2	7 day	7th Sep 2008	depend on processing-time
C3	5 day	3rd Sep 2008	9th Sep 2008

Figure 2. The table shows QoS offered by external companies and their time constraints.

In this illustration, we concentrate on one type of QoS value which is processing time. For time constraint, we identify two kind of times/dates namely Start After/On to show earliest start time that the company can start executing and Finish Before/By to show the latest finish time that the company can participate in the business project.

Assume the process owner wants a combination of companies that can minimize processing time. Based on this requirement, obviously, B2 and C3 are a good combination. However, from the time constraints perspective, both of them will introduce inconsistency. Based on the start-after date, B2 is expected to start on 3rd September while B2 latest finish time depends on the the processing time. Thus, B2 latest finish time would be 5th September. According to activity ordering, C3 can only be started when B2 finishes although it has earliest start time. By taking the latest finish time of B2, C3 earliest start time would be 5th September. As C3 processing time takes 5 days, its latest finish time will be on 10th September which violates its initial latest finish time. Therefore, both of them are said to be inconsistent.

This means other combination has to be selected. Assume the process owner still B2 to be selected. Thus, C1 or C2 would be the potential consistent combination although their QoS offers are not as good as C3. Assume a deadline is imposed for these two activities to be finished within 8 days from the start time of the first activity. In general solution, their QoS values will be aggregated. Thus, combining B2 and C1 will result in 8 days while B2 and C2 will result in 9 days. As a result, we can easily identify that B2 and C1 would be the right combination.

However, there is a waiting time occurs if we examine from the time constraint perspective. As identified, B2 latest finish time is 5th September. Although C2 supposes to start when B2 finishes, C2 has a time constraint where it can only start on 7th September. This means 6th September is a waiting time which should be counted which results in the total time taken to finish is 9 days. In a conclusion, neither

B2 and C1 nor B2 and C2 could contribute to the right combination that can satisfy the deadline.

Maintaining C3 could be the right option. There are two potential combinations with C3 that can satisfy the deadline, in particular B1 with C3 and B2 with C3. However, B2 with C3 are inconsistent. Therefore, B1 with C3 would be the right combination as compared to other companies.

As a composite plan, another two activities have to be considered to satisfy production deadline namely, getting materials and packaging. Based on activity ordering, selected supplier has to prepare the materials before it can be delivered. Selected packager has to wait for product from the manufacturing and also packaging materials. This means the time constraints of potential selected supplier has to be consistent with the selected transporter. Meanwhile the time constraints of the selected packager have to be consistent with the selected transporter and manufacturer. The combination of all selected companies must satisfy the deadline.

Based on this simplified scenario, it is clear that the time constraints are crucial to the service selection for a composition. Besides QoS aggregation, time constraints have to be consistent among companies and satisfy the global requirement. Therefore, in this paper, we propose an approach to tackling these issues from service selection perspective.

3. Selection Approach

We propose a service selection approach that comprises of two major functionalities; (i) Exploration of process model based on patterns (ii) Evaluation of candidates based on time constraints and objective functions. The first function is essential to guide the construction of the selection plan. The second function is meant to find consistent combination which can satisfy global requirements. By having these functionalities, a service selection strategy can be established. Each of these will be detailed out in the following sub sections.

For the notation, the process model is depicted as a directed graph or just graph $G = (N, A)$. N is a set of activities $(1, 2..., n)$. A is a set of arcs where $A \subseteq N \times N$. $b \in N$ is the first activity and $l \in N$ is the last activity. Furthermore, S is a set of services$(1, 2..., m)$ for each activity in N. In this section, potential services and candidates, combination and solution will be used interchangeably.

3.1. Exploring Process Model

The selection plan is constructed step-by-step based on the exploration of the process model. We propose pattern observation to guide the exploration. Starting from the first activity in the process model, specific pattern is observed until the last activity is reached. When a pattern is found, all candidates involved in the pattern will be evaluated.

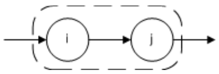


Figure 3. A sequence pattern contains at most two activities per observation

For the pattern, we refer to the work by [2] and concentrate on *sequence* and *parallel* pattern. As there are several types of parallel patterns, we limit the discussion on AND-AND combination(AND-split with AND-merge) and XOR-XOR combination(XOR-split with XOR-merge). These patterns are chosen due to their well-known usage and can be easily aligned to WS-BPEL specification [3].

Based on the process model, several observations can be identified as follows:

- **Observation 1** - This observation is specifically for the first activity, b. Assume b is an operational activity, then the exploration holds for all candidates S_b to be evaluated. As a result, one best candidate is assigned to b.
- **Observation 2** - This observation is meant for an activity and its relation with the suceeding activity. Let $i \in N$ be the current activity that has been assigned with a candidate $x \in S_i$. The exploration observes one succeeding activity of i given as $j \in N$ and i is the only preceding activity for j. Thus a **sequence pattern** is found as illustrated in Fig. 3. All candidates of S_j will be evaluated to find the right combination with $x \in S_i$.
- **Observation 3** - This observation is meant for an activity and its relation with the succeeding activities. Let $i \in N$ be the current activity that has been assigned with a candidate $x \in S_i$. The exploration observes more than one succeeding activity of i given as $(i + 1, ..., i + p) \in N$, then a **parallel pattern**(AND-merge or XOR-merge) is found as illustrated in Fig. 4(Part A). Obviously, the behaviour of these structures i.e., AND-merge and XOR-merge are different during the implementation. However, since we consider the worst case scenario, thus all candidates of $(S_{i+1}, ..., S_{i+p})$ will be evaluated to find the right combination with $x \in S_i$.
- **Observation 4** - This observation is a continuation to the Observation 3. Again, in this observation we are concerned with the worst case scenario and thus we treat equally for both types of parallel structure. Let $(i+ 1, ..., i+p) \in N$ be the current activities that have been assigned with candidates, given as $(a_1 \in S_{i+1}, ..., a_p \in S_{i+p})$. The exploration observes one similar succeeding activity, given as $j \in N$, then a **parallel pattern**(AND-merge or XOR-merge) is found as illustrated in fig.

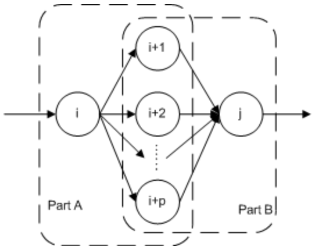


Figure 4. A parallel pattern contains splitting pattern(Part A) and merging pattern(Part B) which are observed separately.

4(Part B). All candidates of S_j will be evaluated to find the right combination with $(a_1 \in S_{i+1}, ..., a_p \in S_{i+p})$.

3.2. Evaluating Candidates

The evaluation of candidates aim at finding a combination that is consistent and meets the global requirements. We adopt the idea of scheduling [4] and some conceptual ideas of constraint-based scheduling [5] to perform consistency check. The evaluation attempts to generate a schedule that subjects to precedence relations, offered QoS value and time constraints. A consistent combination is found when a time-feasible schedule can be achieved. All consistent candidates are further evaluated for the objective function satisfaction to determine whether the global requirement is met or not. A combination that satisfies the global requirement is taken as the feasible selection plan. In this section, we divide the discussion into three sub sections namely scheduling generation, time constraints consistency and objective function satisfaction.

3.2.1. Schedule Generation. The schedule targets to assign two time points for each candidate which are referred as earliest-start-time, given as est and earliest-finish-time, given as eft. As a composite plan, the schedule is subjected to the precedence relationships among candidates. Therefore, for sequence pattern as in Observation 2, the schedule is generated by enforcing precedence relation based on (1). For parallel pattern as in Observation 3, the schedule is generated by enforcing precedence relation based on (2). Meanwhile, for parallel pattern as in Observation 4, the schedule is generated by enforcing precedence relation based on (3).

$$eft_i(x) \leq est_j(y) \qquad \text{Sequential} \quad (1)$$
$$eft_i(x) \leq (est_{i+1}(a_1), ..., est_{i+p}(a_p)) \qquad \text{Splitting} \quad (2)$$
$$(eft_{i+1}(a_1), ..., eft_{i+p}(a_p)) \leq est_j(y) \qquad \text{Merging} \quad (3)$$

The schedule is computed based on the offered QoS values namely processing time, given as q. For enforcing Relation (1), let $y \in S_j$ be the candidate to be scheduled, $est_j(y)$ is computed as in (4). Equation (4) explains that $est_j(y)$ will take the earliest finish time value of the preceding candidate if the current time less than the earliest finish time of the preceding candidate. Otherwise it will take Now as the current time. Equation (4) is also applied to enforce Relation (2). Here we assume each candidate $(est_{i+1}(a_1), ..., est_{i+p}(a_p))$ is depicted as $est_j(y)$. However, for enforcing Relation (3), let $y \in S_j$ be the candidate to be scheduled, it has to wait until all preceding candidates have finished as illustrated in Fig. 5. Thus $est_j(y)$ is computed as in (5). However, if the current time is greater than any of the preceding candidates, then it will take Now value. Meanwhile, eft for any candidates are computed as in (6).

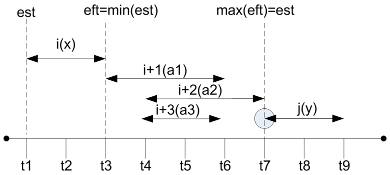


Figure 5. $eft_{i+2}(a_2)$ set the value for $est_j(y)$ since $eft_{i+2}(a_2)$ takes the maximum earliest-finish time

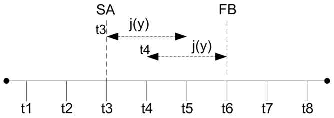


Figure 6. SA sets the earliest start time point and FB sets the latest finish time point. With processing time, given as $q = 2$, the candidate $j(y)$ may start at t_3 or t_4.

$$est_j(y) = \max(eft_i(x), Now) \qquad (4)$$
$$est_j(y) = \max(\max(eft_{i+1}(a_1), ..., eft_{i+p}(a_p)), Now) \qquad (5)$$
$$eft_i(y) = est_i(y) + q_i(y) \qquad (6)$$

3.2.2. Time Constraints Consistency. Time constraints are additional and fixed time points that control the processing time of a candidate. We propose two types of time points namely *start-after-time* given as SA and *finish-before-time* given as FB. SA defines the earliest start time(release time) while FB controls the latest finish time(deadline). Figure 6 illustrates these specialized times constraints.

In the figure, we assume the offered processing time is $q_j(y) = 2$. In reality, SA and FB may take a value such as 5th Sep. However, for the sake of computation, both will be converted to a computed time value e.g., $SA_j(y) = 3$ and $FB_j(y) = 6$. Without considering other candidates, this candidate may be scheduled to start at $est_j(y) = 3$ or $est_j(y) = 4$. However, as a composite plan, this candidate has to be consistent with others. Thus, the schedule of est may takes any of these two decisions:

1) **Maintain** with the computed values as in (4) or (5).
2) **Adjust** the earliest-start-time by taking SA value.

For enforcing Relation (1) and (2), the schedule is formulated based on (4) if $SA_j(y) \leq eft_i(x)$ or based on (7) if $SA_j(y) > eft_i(x)$. For enforcing Relation (3), the schedule is formulated based on (5) if $SA_j(y) \leq max(eft_{i+1}(a_1), ..., eft_{i+p}(a_p))$ or based on (7) if $SA_j(y) > max(eft_{i+1}(a_1), ..., eft_{i+p}(a_p))$.

$$est_j(y) = SA_j(y) \tag{7}$$

It is important to check with the FB (if exists) when maintaining or adjusting earliest-start-time value. Let $y \in S_j$ be any candidate that has been adjusted/maintained, the following rule is applied:

$$(est_j(y) + q_j(y)) \leq FB_j(y)) \tag{8}$$

If (8) is satisfied, then candidate $y \in S_j$ is said to be consistent with precedence relations. This also means a consistent combination is found and a time-feasible schedule is achieved.

3.2.3. Objective Functions Satisfaction.

An objective function can be categorized into two; (i) to achieve optimal solution, (ii) to achieve feasible solution. An optimal solution is achieved when the objective function is optimized i.e., maximized or minimized. In contrast, a feasible solution is achieved when the objective function is satisfied. Furthermore, an objective function can be used to support two level of decisions; local decision and global decision. Local decision involves selection of the best candidate from the same set. Meanwhile global decision involves selection of the entire plan. In this paper, we propose an optimal solution for the local decision and a feasible solution for the global decision. Obviously, the main goal is to find a feasible plan. The feasible plan only considers candidates that are consistent. Thus, we assume that there are two objectives; (i) to complete within certain finish time such as "finish time $\leq$ 10 days" and (ii) to complete within certain cost or price such as " total price $\leq$ \$1000". The first objective is represented by FT and the second objective is represented by PR.

We propose two major procedures to achieve the feasible plan which are as follows:

1) **Ranking** - This procedure is meant to support the local decision. Technically, the best value which refers to the candidate at the top rank is searched. As we are interested in two objectives, both quality values i.e., time and price have to be considered. To obtain the rank value of each candidate, we apply a technique called Simple Additive Technique [6]. In general, this technique will normalize a quality value into a value in the range of 0 to 1. Then, the normalized value will be summed up to get the total value. Let $r_i(x)$ be the ranked value of a candidate, the following formula is applied:

$$r_i(x) = \sum_{u=1}^{2} (\frac{\max V_{iu} - v_{iu}(x)}{\max V_{iu} - \min V_{iu}}) \tag{9}$$

Equation (9) contains u that represents two quality metrics; $u = 1$ refers to the earliest-finish-time and $u = 2$ refers to the price. It also contains v that represents the value referred by u. This means v takes earliest-finish-time value when $u = 1$ and v takes price value when $u = 2$. In addition, V represents all values according to u. Once the rank value is computed for each candidate in the same set, the best candidate/top rank is obtained as follows:

$$top_i = \max_{k=1}^{m}(r_i(k)) \tag{10}$$

In a situation where there is more than one candidate at the top rank, a random selection is implemented.

2) **Aggregation** - This procedure is meant to support global decision. Technically, the value of all selected candidates are aggregated and compared to the objective value. To determine the first objective(related to the finish time), the aggregation takes the computed time into consideration. However, the computation will only be executed after each activity has been assigned with a candidate. The aggregation can be formulated as one of scheduling criteria namely flow-time. A flow-time refers to the total time spent in the entire process. Let F be the flow-time, the formula is defined as follows:

$$F = eft_l(y) - est_b(x) \tag{11}$$

Equation (11) takes a difference between selected candidate of the last activity and the first activity. By formulating this way, any unintended waiting time caused by SA value will be taken into consideration. For the second objective(related to the price), the aggregation is executed during the construction of the plan. This means according to the pattern exploration. Let *pr* be the offered price value, the price aggregation is formulated based on (12) for the sequence pattern as in Observation 2. For the parallel pattern as in Observation 3, the price aggregation is formulated based on (13). Meanwhile, (14) is applied for parallel pattern as in Observation 4.

$$pr_i(x) + pr_j(y) \quad (12)$$

$$pr_i(x) + max(pr_{i+1}(a_1), ..., pr_{i+p}(a_p)) \quad (13)$$

$$max(pr_{i+1}(a_1), ..., pr_{i+p}(a_p)) + pr_j(y) \quad (14)$$

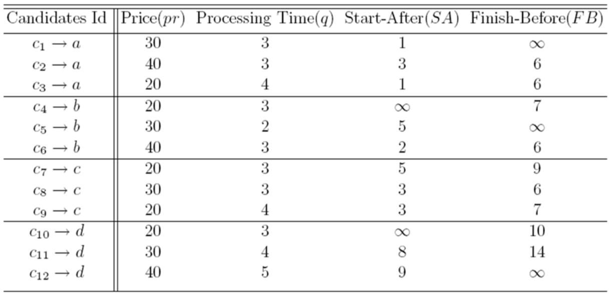
By aggregating these values, a total price given as P can be obtained. When both F and P have been computed, they are checked with the objective value. Thus, F and P are said to be satisfied with the objective function if the following rule results in true:

$$(F \leq FT) \wedge (P \leq PR) \quad (15)$$

3.3. Selection Strategy

The selection strategy requires a few inputs in particular (i) a process that contains activities and their relations, (ii) a set of candidates that map to each activity, (iii) a set of QoS values i.e., processing time, price, (iv) a set of time constraint values i.e., start-after-time and finish-before-time and (v) a set of objective values. The values (iii) and (iv) are assumed to be retrieved from the providers and (v) are retrieved from the user. By having these inputs, the strategy can be implemented as follows:

1) If it is the first activity as in Observation 1, then do the following:
   a) Compute est for each candidate as in (4) or (7). Then compute for eft as in (6). While computing, check with FB as in (8).
   b) If none of the candidates are consistent, then return failure.
   c) For each consistent candidate, rank them as in (9) and select one candidate as in (10).
2) If it is a sequence pattern as in Observation 2, then do the following:
   a) Compute est for each candidate as in (4) or (7). Then compute for eft as in (6). While computing, check with FB as in (8).
   b) If none of the candidates are consistent, then return failure.
   c) For each consistent candidate, rank them as in (9) and select one candidate as in (10).
   d) Aggregate price value as in (12).
3) If it is a parallel pattern as in Observation 3, then do the following:
   a) Compute est for each candidate as in (4) or (7). Then compute for eft as in (6). While computing, check with FB as in (8).
   b) If none of the candidates are consistent, then return failure.
   c) For each consistent candidate, rank them as in (9) and select one candidate as in (10).
   d) Aggregate price value as in (13).
4) If it is a parallel pattern as in Observation 4, then do the following:
   a) Compute est for each candidate as in (5) or (7). Then compute for eft as in (6). While computing, check with FB as in (8).
   b) If none of the candidates are consistent, then return failure.
   c) For each consistent candidate, rank them as in (9) and select one candidate as in (10).
   d) Aggregate price value as in (14).
5) Repeat step 2, 3, and 4 until all activities have been explored and assigned with a candidate.
6) Aggregate the flow-time as in (11).
7) Check with the objective functions as in (15).



Candidates Id	Price(pr)	Processing Time(q)	Start-After(SA)	Finish-Before(FB)
$c_1 \rightarrow a$	30	3	1	∞
$c_2 \rightarrow a$	40	3	3	6
$c_3 \rightarrow a$	20	4	1	6
$c_4 \rightarrow b$	20	3	∞	7
$c_5 \rightarrow b$	30	2	5	∞
$c_6 \rightarrow b$	40	3	2	6
$c_7 \rightarrow c$	20	3	5	9
$c_8 \rightarrow c$	30	3	3	6
$c_9 \rightarrow c$	20	4	3	7
$c_{10} \rightarrow d$	20	3	∞	10
$c_{11} \rightarrow d$	30	4	8	14
$c_{12} \rightarrow d$	40	5	9	∞

Figure 7. The table shows potential candidates for activities in parallel structure. Each candidate has two quality values and two potential time constraints. The $\rightarrow$ symbol shows which activity that a candidate belong to while ∞ symbol shows the value is not given.

4. Case Study

This case study is meant to illustrate the proposed selection approach. For simplicity, we only focus on the *parallel* pattern. For the sake of illustration, let a be the splitting activity(also assume as the first activity), b and c be the parallel activities and d be the merging activity(also assume as the last activity). Each activity has three potential candidates as presented in Fig. 7. The outcomes of this case study are presented in Fig. 9.

As proposed in the approach, activity a matches Observation 1 as the first activity. Thus, each candidate for a are scheduled as in (4) or (7) and followed by (6). The generated schedules are checked with FB as in (8) which results in consistent for all candidates. Then, each candidate is ranked based on (9) and one candidate is selected based on (10) which takes c_1 as selected candidate.

The solution is constructed step-by-step. Thus, splitting activities are treated first as in Observation 3. Each candidate for b and c are scheduled as in (4) or (7) which takes c_1 as the preceding candidate as illustrated in Fig. 8(A). Then

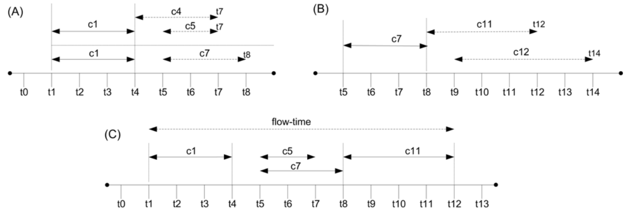
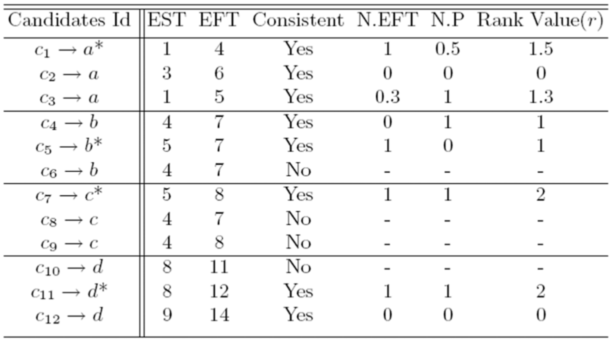


Figure 8. Figure (A) shows all potential candidates that are consistent with c_1. Figure (B) shows all potential candidates that are consistent with c_7. Figure (C) shows all selected candidates which contribute to the computation of flow-time.



Candidates Id	EST	EFT	Consistent	N.EFT	N.P	Rank Value(r)
$c_1 \rightarrow a*$	1	4	Yes	1	0.5	1.5
$c_2 \rightarrow a$	3	6	Yes	0	0	0
$c_3 \rightarrow a$	1	5	Yes	0.3	1	1.3
$c_4 \rightarrow b$	4	7	Yes	0	1	1
$c_5 \rightarrow b*$	5	7	Yes	1	0	1
$c_6 \rightarrow b$	4	7	No	-	-	-
$c_7 \rightarrow c*$	5	8	Yes	1	1	2
$c_8 \rightarrow c$	4	7	No	-	-	-
$c_9 \rightarrow c$	4	8	No	-	-	-
$c_{10} \rightarrow d$	8	11	No	-	-	-
$c_{11} \rightarrow d*$	8	12	Yes	1	1	2
$c_{12} \rightarrow d$	9	14	Yes	0	0	0

Figure 9. The table shows all relevant results based on the proposed approach. Note that EST stands for earliest-start-time and EFT refers to earliest-finish-time. N.EFT represents normalized value of earliest-finish-time. N.P represents Normalized value for price. The symbol '*' shows selected candidates.

it is followed by eft value as in (6) while checking for consistency as in (8). For candidates of activity b, the results show that c_6 is inconsistent while others are consistent. All consistent candidates are ranked as in (9). Note that, both c_4 and c_5 have similar rank value. In this case, we assume random selection is made which results in candidate c_5. Meanwhile, consistency check for candidates for c results in only c_7 as the consistent candidate. Obviously c_7 is selected for activity c. Once all activities have been assigned with a candidate, the aggregation of price is computed as in (13).

The solution then moves to the merging pattern as in Observation 4. Each candidate is scheduled as in (5) or (7). For the preceding candidate, the formula takes candidate the finish time of c_7 as illustrated in Fig. 8(B). Consistency check based on (8) confirms that c_{11} and c_{12} are consistent. Both are then ranked as in (9) and c_{11} is selected since it satisfies (10). Again, once all activities have been assigned with a candidate, the aggregation of price is computed as in (14).

Once the assignment to all activities is complete, given as $c_1 \rightarrow a$, $c_5 \rightarrow b$, $c_7 \rightarrow c$, $c_{11} \rightarrow d$, the flow-time as illustrated in Fig. 8(C) can be computed according to (11). The aggregated values result in $F = 11$ and $P = 90$. Both values are then checked with the objective value as in (15). A complete solution is found if the rule is satisfied.

5. Related Works

Service selection problems have recently attracted researchers' attention. One of the main goals is to select a group of services that can satisfy the global requirements. QoS values are the crucial inputs to support service selection for service composition. Several QoS computational models have been proposed which can be categorized into two approaches. Firstly, QoS computational model based on the composition structures which are explored from a process. In this model the computation can be done at any region in a process based on specific structure or pattern. In [10], a basic QoS model which computes execution time and cost is presented. The computation involves two main structures namely sequential and parallel structure. A comprehensive QoS Model is presented in [7] and [9] which covers more QoS metrics and structures. Secondly, QoS computational model based on the execution paths which are explored from a process. In this model, the compositional value can be obtained easily by focusing on specific path. To avoid exhaustive computation on a huge number of paths, the work by [8] has focused on the computational of critical path. In regard to our work, we follow the direction of QoS model i.e., processing time and price based on the composition structures or workflow patterns. However, due to the temporal constraints issue, we consider the exploration from the first activity to the last activity. Furthermore, we take different approach for computing the flow time to ensure that any unintended waiting time is taken into consideration.

Existing approaches on the selection commonly work on two levels, local and global selection. In [8] [12], they used Simple Additive Weighting (SAW) technique to find the best candidate in the same set for the local selection. Furthermore, in [8] [11] [13], they transformed the problem into global optimization for the global selection. Meanwhile the work by [12] has suggested a pattern-wise selection as a graph-based approach to apply for the global selection. In our work, we also apply the local and global selection. For the local selection, we use SAW technique by considering input based on the consistency checking. Meanwhile for the global selection, we focus on finding a feasible solution according to graph-based approach. Our strategies have some similarities with the suggestion given by [12] but we provide more detail steps with an additional issue.

Temporal constraint issues have been investigated in many areas and its play an important role in reasoning about time as presented comprehensively in [14] [15]. One of the reasoning issues is to ensure the consistency of time constraint.

However, this issue has not been sufficiently investigated in the area of service selection. The works by [16] have a close relation to our work. The temporal constraints are used to define validity period of the requirements and capabilities of web services in procurement. In this context, the consistency checking is applied to ensure that no overlapping between temporal constraints of different requirements or capabilities. In regard to the service selection, we consider an additional perspective in checking for the consistency namely the ordering constraints e.g., sequential, parallel between candidates. By considering multiple perspectives, it requires for more specific consistency rules to identify consistent candidates to be participated in the selection plan.

6. Conclusion

Time constraints bring about a great impact to the service selection especially when dealing with collaborative environment. The selection plan should be provided with time constraints consistency check capability to ensure consistent combination between selected services while maintaining precedence relationships and also satisfying the objective functions. In this work, we focus on the feasible solution where the best value is selected and aggregated to form a solution. The exploration of the process model is guided based on patterns.

There are lots of works that can be done in addressing this issue. One of the promising aspects is when consistency rule is violated where specific computation can be applied to understand the degree of violation. Instead of simply arising a failure notification, the result of the computation can be useful in determining further actions.

Our future aim is to tackle run-time situation where changes may occur and reselection is required. Obviously, addressing this issue contributes to major significance for business process implementation using web service technology.

Acknowledgment

This research is partly supported by the Australian Research Council Discovery Project Scheme under grant number DP0663841.

References

[1] M. P. Papazoglou. *Web Services: Principles and Technology*. Pearson, Prentice Hall, 2008.

[2] W. M. P. van der Aalst, A. H. M. T. Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.

[3] D. Jordan, J. Evdemon, A. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. Ford, Y. Goland, A. Guizar, N. Kartha, C. K. Liu, R. Khalaf, D. Konig, M. Marin, V. Mehta, S. Thatte, D. v. d. Rijn, P. Yendluri, and A. Yiu. Web services business process execution language version 2.0. OASIS, 2007.

[4] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling*. Dover Publications, Inc., 2003.

[5] P. Baptiste, P. Laborie, C. L. Pape, and W. Nuijten. *Handbook of Constraint Programming*, chapter Constraint-Based Scheduling and Planning, pages 761–799. Elsevier, 2006.

[6] H.C.-L and K. Yoon. *Multiple Criteria Decision Making*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1981.

[7] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):281–308, 2004.

[8] L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, 30(5):311–327, 2004.

[9] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl. Qos aggregation for web service composition using workflow patterns. In *Proceedings of Eighth IEEE International Enterprise Distributed Object Computing Conference*, pages 149–159, 2004.

[10] D. A. Menasce. Composing web services: A qos view. *IEEE Internet Computing*, 8(6):88–90, 2004.

[11] D. Ardagna and B. Pernici. Global and local qos guarantee in web service selection. In *Business Process Management Workshops*, pages 32–46, 2006.

[12] R. Gronmo and M. C. Jaeger. Model-driven methodology for building qos-optimised web service compositions. In *Distributed Applications and Interoperable Systems*, pages 68–82. Springer Press, 2005.

[13] S. Liu, Y. Liu, N. Jing, G. Tang, and Y. Tang. A dynamic web service selection strategy with qos global optimization based on multi-objective genetic algorithm. In *Grid and Cooperative Computing - GCC 2005*, pages 84–89, 2005.

[14] L. Chittaro and A. Montanari. Temporal representation and reasoning in artificial intelligence: Issues and approaches. *Annals of Mathematics and Artificial Intelligence*, 28(1):47–106, 2000.

[15] M. Koubarakis. *Handbook of Constraint Programming*, chapter Temporal CSPs, pages 665–697. Elsevier, 2006.

[16] O. Martin-Diaz, A. Ruiz-Cortes, A. Duran, and C. Muller. An approach to temporal-aware procurement of web services. In *Service-Oriented Computing - ICSOC 2005*, pages 170–184, 2005.