An Approach Based on Fuzzy Sets to Handle Preferences in Service Retrieval

Katia Abbaci, Fernando Lemos, Allel Hadjali, Daniela Grigori, Ludovic Lietard, Daniel Rocacher, Mokrane Bouzeghoub

HAL Id: hal-00672480
https://hal.science/hal-00672480

Submitted on 21 Feb 2012

An Approach Based on Fuzzy Sets to Selecting and Ranking Business Processes

Katia Abbaci, Fernando Lemos, Allel Hadjali, Daniela Grigori, Ludovic Liétard, Daniel Rocacher, Mokrane Bouzeghoub

Abstract—Current approaches for service discovery are based on semantic knowledge, such as ontologies and service behavior (described as a process model). However, these approaches have high selectivity rate, resulting in a large number of services offering similar functionalities and behavior. One way to improve the selectivity rate is to cope with user preferences defined on quality attributes. In this paper, we propose a novel approach for service retrieval that takes into account the service process model and relies both on preference satisfiability and structural similarity. User query and target process models are represented as annotated graphs, where user preferences on QoS attributes are modelled by means of fuzzy sets. A flexible evaluation strategy based on fuzzy linguistic quantifiers is introduced. Finally, different ranking methods are discussed.

Index Terms—web service retrieval, quality of services, preferences, fuzzy set theory, linguistic quantifier

I. Introduction

Searching for a specific service within service repositories become a critical issue for the success of service oriented and model-driven architectures and for service computing in general. This issue has recently received considerable attention and many approaches have been proposed. Most of them are based on the matchmaking of process input/outputs [1], service behavior (described as process model) [2], [3] or ontological knowledge [3]. However, these approaches have high selectivity rate, resulting in a large number of services offering similar functionalities and behavior [3].

One way to discriminate between similar services is to consider non-functional requirements such as quality preferences (response time, availability, etc.). A recent trend towards quality-aware approaches has been initiated [4], [5], but it is limited to atomic services. Our goal is to go further these approaches into a unique integrated approach dealing with functional and non-functional requirements in service retrieval. Targeting this goal poses the following two challenges: (i) At the description level, provide a model allowing to specify non-functional requirements at different granularity levels of the service functional description; (ii) At the discovery level, define an evaluation method that efficiently computes the satisfiability of a target service w.r.t. the functional and non-functional requirements of a user query.

More specific challenges related to non-functional characteristics should also be taken into account: (i) Users are not always able to precisely specify their non-functional constraints; (ii) Users have different points of view over what is a satisfactory service according to the same set of non-functional

constraints; (iii) The service retrieval should avoid empty or overloaded answers due to the imprecision of the user's query.

Preferences are a natural way to facilitate the definition of non-functional constraints in user query. They are flexible enough, on the one hand, to avoid empty returns caused by very strict user constrains and, on the other hand, to provide an adequate set of relevant results even when user specifies too general constraints. In addition, fuzzy logic has been used as a key technique to take into account human point of view in preference modelling and evaluations [6].

In [7], a QoS-aware process discovery method is proposed whereas user query is a graph annotated with QoS factors. Starting from [7], this paper investigates a novel approach for services selection and ranking taking into account both behavior specification and QoS preferences. User query and target process models are represented as graphs, where queries are annotated with preferences on QoS properties and targets are annotated with QoS attributes. Preferences are represented by means of fuzzy sets as they are more suitable to the interpretation of linguistic terms (such as *high* or *fast*) that constitute a convenient way for users to express their preferences. To avoid empty answers for a query, an appropriate flexible evaluation strategy based on fuzzy linguistic quantifiers (such as *almost all*) is introduced.

In the remainder of this paper, Section II provides some basic background. Section III describes process model specification with preferences. Section IV addresses fuzzy preference modelling and evaluation. Section V presents our interpretation of process models similarity based on linguistic quantifiers. In Section VI, service ranking methods are discussed. Section VII proposes an illustrative example.

II. Background and Related Work

In this section, we first recall some necessary notions on preference modelling. Next, we review preference-based service discovery approaches.

A. Preference Modelling

The semantics of preferences assumed in this work is the one provided by the databases area: preferences are used to reduce the amount of information returned as response to user queries and to avoid the empty answers. Generally, two families of approaches can be distinguished to model preferences. The first family gathers approaches that rely on commensurability assumption which leads to a total pre-order

[8]. The second one comprises approaches that assume that commensurability does not hold, in this case no compensation on is allowed between criteria and only a partial order is obtained [9].

Fuzzy sets were introduced by Zadeh [10] for dealing with the representation of classes or sets whose boundaries are not well defined. Then, there is a gradual rather than crisp transition between the full membership and the full mismatch. Typical examples of such fuzzy classes are those described using adjectives of the natural language, such as *cheap*, *fast*, etc. Formally, a fuzzy set F on the universe X is described by a membership function $\mu_F : X \rightarrow [0, 1]$, where $\mu_F(x)$ represents the **membership degree** of x in F. By definition, if $\mu_F(x) = 0$ then the element x **does not belong at all** to the fuzzy set F, if $\mu_F(x) = 1$ then x **fully belongs** to F. When $0 < \mu_F(x) < 1$, one speaks of **partial membership**. The set $\{x \in F | \mu_F(x) > 0\}$ represents the **support** of F and the set $\{x \in F | \mu_F(x) = 1\}$ represents its **core**.

The membership function associated to F is often represented by a trapezoid $(\alpha, \beta, \varphi, \psi)$[1], where $[\alpha, \psi]$ is its support and $[\beta, \varphi]$ is its core. Among other forms (Gaussian, sigmoidal, ...), this one is easy to be defined and to manipulate.

B. Preference-based Service Discovery

Most of the first approaches for service discovery using preferences were based on crisp logic solution and considered the services as black boxes [5]. With regard to the specification model, some of them do not deal with preferences [11]. The other approaches does not propose or use preference constructors to help user better define his/her preferences or interpret the results [5], [12].

The existing fuzzy approaches [13], [4] take into account only the satisfiability of preferences whereas they ignore the structural similarity of web services. In addition, most of them do not verify the *subjectivity property*, which considers the user point of view when defining the membership functions. Moreover, these works deal only with services as black boxes. In this paper, user can also define preferences over the activities of the service behavior specification. We also propose an approach for service selection where both structural similarity and preference satisfiability are considered.

III. Preferences in Process Model Specification

Many languages are available to describe service process models, e.g., BPEL4WS and OWL-S. They represent a process model as a set of atomic activities combined using control flow structures. As a consequence, these languages can be abstracted as a direct graph $G = (V, E)$, where the vertices represent activities (e.g., hotel reservation, shipping user preferences, payment) or control flow nodes (e.g., and, or, etc.), while the edges represent the flow of execution (e.g, the edge between the two activity nodes, hotel reservation and payment, means that these two activities run in a sequential order).

In this work, services are specified as graphs annotated with QoS properties and user queries are specified as graphs

annotated with preferences. Figure 1 shows an example of a process model annotated with QoS attributes. The example presents a global annotation indicating the security of the process model and activity annotations indicating the response time, reliability and cost of some activities. Figure 2 shows a sample user query annotated with a global preference indicating user prefers services providing RSA encryption and some activity preferences over reliability, response time and cost.

We do not discuss here the techniques to obtain the QoS information of a process model. For this, consider the work in [14]. Next, we present the formal definitions of our model.
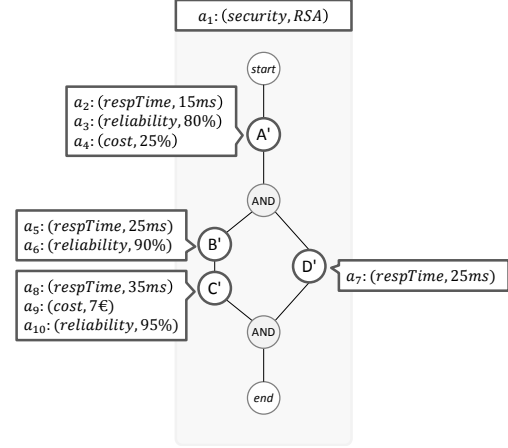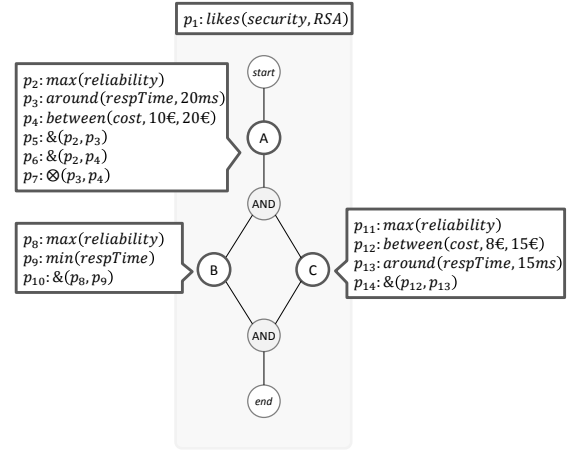

Figure 1. Target Graph t_1

Figure 2. Query Graph q_1

Definition 1. *An **annotation** is a pair (m, r), where m is a QoS attribute obtained from an ontology O and r is a value for m[2]. It can be specified over a process model graph (**global annotation**) or over an atomic activity (**activity annotation**).*

Definition 2. *A **preference** is an expression that represents a desire of the user over the QoS attributes of a process model or activity. It can be of one the following forms[3]:*

---

[1] In our case, $(\alpha, \beta, \varphi, \psi)$ is user-defined to ensure the subjectivity.

[2] We abstract from the different units in which a value can be described.
[3] Based on a subset of preferences defined in [15].

- *atomic preferences:*
  - $around\,(m, r_{desired}, \mu_{around})$: *for attribute m, this expression favors the value $r_{desired}$; otherwise, it favors those close to $r_{desired}$.*
  - $between\,(m, r_{low}, r_{up}, \mu_{between})$: *for attribute m, it favors the values inside the interval $[r_{low}, r_{up}]$; otherwise, it favors the values close to the limits.*
  - $max\,(m, \mu_{max})$: *for attribute m, it favors the highest value; otherwise, the closest value to the maximum is favored. For example, the maximum of availability is equal by default to 100%.*
  - $min\,(m, \mu_{min})$: *for attribute m, it favors the lowest value; otherwise, the closest value to the minimum is favored, as example: the minimum of response time or cost is equal by default to 0.*
  - $likes\,(m, r_{desired})$: *for attribute m, it favors the value $r_{desired}$; otherwise, any other value is accepted;*
  - $dislikes\,(m, r_{undesired})$: *for attribute m, it favors the values that are not equal to $r_{undesired}$; otherwise, $r_{undesired}$ is accepted;*
- *complex preferences:*
  - *Pareto preference $\otimes\,(p_i, p_j)$: this expression states that the two soft preference expressions p_i and p_j are equally important;*
  - *Prioritized preference $\&\,(p_i, p_j)$: this expression states that the soft preference expression p_i is more important than the soft preference expression p_j.*

*It can be specified over a process model graph (**global preference**) or over an atomic activity (**activity preference**).*

IV. A Fuzzy Model to Evaluate Preferences

In this section, we introduce a fuzzy semantics of the atomic preferences discussed in the previous section, and show how they can be evaluated. In particular, we propose a metric, called *satisfiability degree* (δ), that measures how well a set of annotations of a target process model satisfies a set of preferences present in the query. As follows, the computation of this degree is done both for atomic and complex preferences.

A. Atomic Preferences

For numerical atomic preferences (i.e. $around$, $between$, max, min), the satisfiability degree is obtained using the user-specific membership functions. Table I summarizes the fuzzy modelling of numerical preferences of interest. Given a preference p and an annotation $a : (m, r)$, one is interested in computing the degree to which the annotation a satisfies the fuzzy characterization underlying p.

For example, consider the constructor $between$: a fuzzy preference $p : between(m, r_{low}, r_{up})$ is characterized by the membership function $(\alpha, \beta, \varphi, \psi)$, where $\beta = r_{low}$; $\varphi = r_{up}$; α and ψ are two values from the universe X. Let $a : (m, r)$ be an annotation of a target graph, the satisfiability degree of preference p according to a is given by:

- p is completely satisfied iff $r \in [r_{low}, r_{up}]$: $\delta\,(p, a) = 1$;
- more r is lower than r_{low} or higher than r_{up}, less p is satisfied: $0 < \delta\,(p, a) < 1$;

- for $r \in \,]-\infty, \alpha] \cup [\psi, +\infty[$, p is not satisfied: $\delta\,(p, a) = 0$.

For non-numerical atomic preferences (i.e. $likes$, $dislikes$), the satisfiability degree is based on the semantic similarity between concepts. We applied the widely known semantic similarity proposed in [16], which states that given an ontology O and two concepts c_1 and c_2, the semantic similarity wp between c_1 and c_2 is given by $wp\,(O, c_1, c_2) = {2N_3}/{N_1 + N_2 + 2N_3}$, where c_3 is the least common super-concept of c_1 and c_2, N_1 is the length of the path from c_1 to c_3, N_2 is the length of the path from c_2 to c_3, and N_3 is the length of the path from c_3 to the root of the ontology. Given a non-numerical atomic preference p and an annotation a, the satisfiability degree $\delta\,(p, a)$ is given by:

- If $p = likes\,(m, r_{desired})$, then
$$\delta\,(p, a) = \begin{cases} 1, & r_{desired} = r \\ wp(O, r_{desired}, r), & otherwise \end{cases}$$
- If $p = dislikes\,(m, r_{undesired})$, then
$$\delta\,(p, a) = 1 - \delta\,(likes\,(m, r_{undesired})\,, a)$$

B. Complex Preferences

To compute the satisfiability degree of complex preferences, we first construct a *preference tree* t_p that represents the semantics of a set of complex preferences S_p. In that preference tree, the nodes represent atomic preferences and the edges represent a *more important than* relation (*prioritized preference*, denoted by $\&$) from parent to child. Preferences belonging to the same level and having the same parent express *Pareto preference*, denoted by $\otimes$. Each level i of the tree is associated with a weight $\omega_i = {1}/{i}$ except the $level\ 0$.

For example, consider the preference tree of q_1 in Figure 3. Preferences p_{11} is an atomic preference that is not component of any complex preference. $p_5 : \&\,(p_2, p_3)$ is a complex preference composed of atomic preferences p_2 and p_3; it means that p_2 is more important than p_3. $p_7 : \otimes\,(p_3, p_4)$ is a complex preference composed of atomic preferences p_3 and p_4; it means that p_3 and p_4 are equally important.
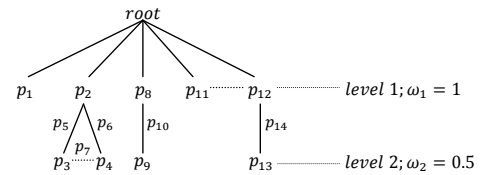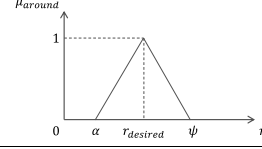

Figure 3. Sample preference tree
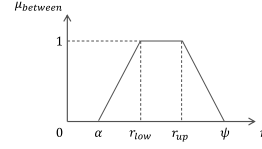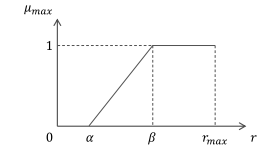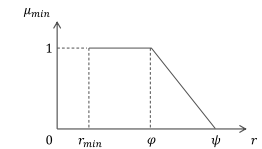
Considering that each atomic preference p_i has a satisfiability degree δ_i, a new satisfiability degree δ'_i is computed taking into account the weight ω_i underlying p_i in the spirit of [6]. δ'_i is defined[4] using the formula (1).

$$\delta'_i = \max\,(\delta_i, 1 - \omega_i) \tag{1}$$

This new interpretation of p_i considers as acceptable any value outside of its support with the degree $1 - \omega_i$. It means

---

[4]We assume here that $\max_{i=1,n} w_i = 1$

Table I
FUZZY MODELLING OF NUMERICAL PREFERENCES

NUMERICAL PREFERENCE	FUZZY INTERPRETATION	
$around(m, r_{desired}, \mu_{around})$	$\mu_{around}(r) = \begin{cases} 0, & r \leq \alpha, r \geq \psi \\ \frac{r-\alpha}{r_{desired}-\alpha}, & \alpha < r < r_{desired} \\ 1, & r = r_{desired} \\ \frac{\psi-r}{\psi-r_{desired}}, & r_{desired} < r < \psi \end{cases}$	
$between(m, r_{low}, r_{up}, \mu_{between})$	$\mu_{between}(r) = \begin{cases} 0, & r \leq \alpha, r \geq \psi \\ \frac{r-\alpha}{r_{low}-\alpha}, & \alpha < r < r_{low} \\ 1, & r_{low} \leq r \leq r_{up} \\ \frac{\psi-r}{\psi-r_{up}}, & r_{up} < r < \psi \end{cases}$	
$max(m, \mu_{max})$	$\mu_{max}(r) = \begin{cases} 0, & r \leq \alpha \\ \frac{r-\alpha}{\beta-\alpha}, & \alpha < r < \beta \\ 1, & \beta \leq r \leq r_{max} \end{cases}$	
$min(m, \mu_{min})$	$\mu_{min}(r) = \begin{cases} 1, & r_{min} \leq r \leq \varphi \\ \frac{\psi-r}{\psi-\varphi}, & \alpha < r < \psi \\ 0, & r \geq \psi \end{cases}$	

that the larger ω_i (i.e., p_i is important), the smaller the degree of acceptability of a value outside the support of p_i. At the end, we have calculated the satisfiability degree of user atomic preferences considering their constructors and the complex preferences composing them.

V. PROCESS MODEL SIMILARITY: A LINGUISTIC QUANTIFIER-BASED METHOD

We describe here a method to compute preference satisfiability between process model graphs. We also discuss a method to assess the structural similarity between two process model graphs. Both degrees will be used to rank potential targets (see Section VI). We precise that this work is not interested in discovering a mapping between two process models; we suppose a mapping already exists such that we can compare matched activities annotations against user preferences. In this issue, please consider the work in [3] for an algorithm that returns a mapping between two process models.

To evaluate the structural similarity of two graphs q and t, we propose to use a graph matching algorithm like in [3]. This algorithm returns a mapping M and a set E of edit operations necessary to transform q into t. A mapping between q and t is a set of pairs (v, w), such that v is an activity of q and w is an activity of t. The edit operations considered are simple graph edit operations: node/edge deletion, addition and substitution. Figure 4 illustrates a mapping between query graph q_1 and target graph t_1. Let $SS(v, w)$ denotes the structural similarity between activities v and w; we use the metric proposed in [3]. Let $\delta(q_1.S_p, t_1.S_a)$ be the satisfiability degree between global

preferences and annotations and let $\delta(v, w)$ be the satisfiability degree between activities v and w (see Section IV).
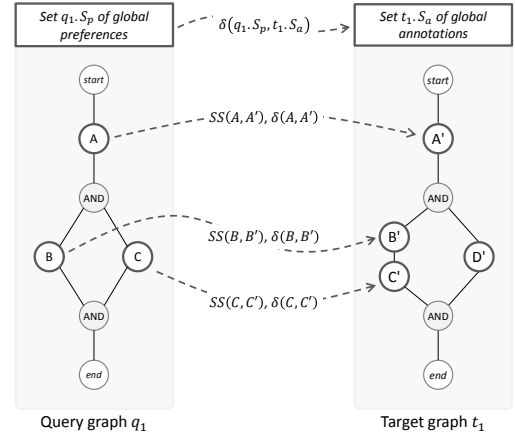
Figure 4. Sample mapping M between query graph q_1 and target graph t_1

Next, we rely on the linguistic quantifier *"almost all"* for the similarity evaluation process. This quantifier is a relaxation of the universal quantifier *"all"* and constitutes an appropriate tool to avoid empty answers since it retrieves elements that would not be selected when using the quantifier *"all"*.

A. Preference Satisfiability between Process Models

A natural user interpretation of the similarity between query and target process models according to user preferences is given by the truth degree of the following proposition:

γ_1: Almost all preferences of q are satisfied by t

The above statement is a fuzzy quantified proposition of the form "$Q\,X\,are\,P$", where (i) Q is a relative quantifier (e.g., *almost all*, *around half*, etc.) which is defined by a function μ_Q such as $\mu_Q(\varpi)$ is the degree of truth of "$Q\,X\,are\,P$" when a proportion ϖ of elements of X fully satisfy A and the other elements being not satisfied; (ii) X is a set of elements; (iii) P is a fuzzy predicate. In [17], a decomposition method to compute the truth degree δ_γ of $\gamma : Q\,X\,are\,P$ is proposed. The method is a two-step procedure:

- Let $\Omega = \{\mu_1, \ldots, \mu_n\}$ be a set of degrees of elements of X w.r.t. P, in decreasing order; i.e. $\mu_1 \geq \ldots \geq \mu_n$;
- The truth degree δ_γ is given by the equation (2), where $\mu_Q(i/n)$ is a membership degree of the element i/n to Q.

$$\delta_\gamma = \max_{1 \leq i \leq n} \min(\mu_i, \mu_Q(i/n)) \qquad (2)$$

In our case, $\Omega = \left\{\mu_1 : \delta_1', \ldots, \mu_n : \delta_n'\right\}$ is the set of satisfiability degrees of all atomic preferences (i.e. global and activity atomic preferences) of query q, where δ_i' is the satisfiability degree of atomic preference p_i computed by (1).

B. Structural Similarity between Process Models

Similarly, we can apply the technique of fuzzy quantifiers to obtain a structural similarity degree between two process models. The structural similarity between a query and target process models can be given by the truth degree of the following propositions:

γ_2: Almost all the activities of q are mapped with activities of t, and

γ_3: Almost no edit operation is necessary to transform q into t

The truth degree of proposition γ_2 is obtained from the formula (2), where $\Omega = \{\mu_1 : SS_1, \ldots, \mu_n : SS_n\}$ is the set of semantic similarity degrees of all mapped activities of q, and SS_i is the semantic similarity degree of a query activity v mapped with a target activity w. In the case of the proposition γ_3, the expression "almost no edit operation is necessary to transform q into t" is equivalent to the expression "*almost all* edit operations are *not* necessary to transform q into t". Therefore, its truth degree is computed as follows:

$$\delta_\gamma = \max_{1 \leq i \leq n} \min(1 - \mu_i, 1 - \mu_Q(i/n)) \qquad (3)$$

In this case, $\Omega = \{\mu_1 : C_1, \ldots, \mu_n : C_n\}$ is the set of transformation costs of mapped target activities with the corresponding activities of q, and C_i is the transformation cost of a target activity w into a query activity v. So, the structural similarity between q and t is evaluated as follows:

$$SS = \min(\delta_{\gamma_2}, \delta_{\gamma_3}) \qquad (4)$$

In our approach, we consider particularly the formulae (2) and (3), where $\mu_Q(i/n) = i/n$. Thus, the meaning of delivered degrees has a simple and clear semantics for the user [18]. For instance, the evaluation of γ_1, γ_2 and γ_3 means that:

"At least $\delta_{\gamma_1}^*\%$ of preferences of q are satisfied by t to at least a degree of δ_{γ_1}, at least $\delta_{\gamma_2}^*\%$ of the activities of q are mapped

with t to at least a degree of δ_{γ_2}, and at least $\delta_{\gamma_3}^*\%$ of q's structure does not need edit operation to transform q into t to at least a degree of δ_{γ_3}" (where $\delta_{\gamma_i}^* = 100 \times \delta_{\gamma_i}$).

VI. PROCESS MODEL RANKING

Previous section presented a fuzzy set-based approach to compute the similarity between one query and one target graph. In this section, given a set of target graphs that are relevant to the query, we discuss some methods to rank-order these graphs according to their structural and preference similarities. Let $\delta(q, t, M)$ be the satisfiability degree between query graph q and target graph t according to a mapping M. Similarly, let $SS(q, t, M, E)$ be the structural similarity between q and t according to a mapping M and a set E of edit operations. We classify ranking methods into two categories:

Ranking Methods based on Aggregation : In this first category, ranking methods aggregate both structural and preference similarities into a unique degree used to rank-order the target graphs. Two kind of aggregations are considered:

Weighted Average-Based Aggregation: $rank(q, t) = \omega_{SS} \times SS(q, t, M, E) + (1 - \omega_{SS}) \times \delta(q, t, M)$ where $0 < \omega_{SS} < 1$ is a weight assigned to the structural similarity criterion.

Min-Combination Based Aggregation: $rank(q, t) = \min(SS(q, t, M, E), \delta(q, t, M))$

Ranking Method without Aggregation : The answers are ranked by using the *lexicographic order*. A priority is given to the structural similarity while the preference similarity is only used to break ties.

VII. ILLUSTRATIVE EXAMPLE

We give here an example of service discovery for query q_1 of Figure 2. We consider a set $\{t_1, \ldots, t_5\}$ of five potential answers to q_1 retrieved by a matchmaking algorithm as discussed in Section V. First, we compute the preference satisfiability between q_1 and the potential target graphs (see Section V-A). Next, we compute the structural similarity between q_1 and the potential targets (see Section V-B). Then, we apply the ranking methods described in Section VI. To illustrate, we evaluate the preference satisfiability and structural similarity between q_1 and target t_1 of Figure 1. We consider the mapping between them as depicted in Figure 4.

Preferences Satisfiability. First, the satisfiability degree δ_i' of each preference p_i of q_1 is calculated. For instance, the satisfiability degree $\delta_2 = \delta(p_2, a_2)$ between preference p_2 and annotation a_2 is obtained by function $\mu_{max}[reliability]$. According to equation (1) and the generated preference tree (Figure 3), the new interpretation of the satisfiability degrees is presented as δ_i'. Depending on the membership function defined for each preference of q_1 and its weight providing by preference tree of Figure 3, satisfiability degrees between query preferences and target annotations are as follows: $\delta_1' = \delta_2' = \delta_8' = \delta_{11}' = 1$, $\delta_3' = \delta_4' = \delta_{13}' = 0.5$, $\delta_9' = 0.9$ and $\delta_{12}' = 0.75$. Second, we apply the truth degree described in Section V-A to obtain the global satisfiability degree between q_1 and t_1, as follows: $\delta_{\gamma_1}(q_1, t_1)$

Table II

STRUCTURAL SIMILARITY AND PREFERENCE SATISFIABILITY DEGREES

TARGET GRAPH	STRUCTURAL SIMILARITY SS	SATISFIABILITY DEGREE δ
t_1	0.66	0.67
t_2	0.29	0.72
t_3	0.85	0.40
t_4	0.78	0.21
t_5	0.68	0.72

Table III

RANKING OF TARGET GRAPHS

WEIGHTED AVERAGE		MIN-COMBINATION		LEXICOGRAPHIC ORDER
t_3	$wa = 0.74$	t_5	$mc = 0.68$	t_3
t_5	$wa = 0.69$	t_1	$mc = 0.66$	t_4
t_1	$wa = 0.66$	t_3	$mc = 0.40$	t_5
t_4	$wa = 0.64$	t_2	$mc = 0.29$	t_1
t_2	$wa = 0.40$	t_4	$mc = 0.21$	t_2

$= \max(\min(1, \mu_Q(1/9)),...,\min(0.5, \mu_Q(9/9))) = 0.67$. This means that at least 67% of preferences of q_1 are satisfied by t_1 to at least a degree 0.67.

Structural Similarity. Assume now that the structural similarities between activities are given by $SS(A, A') = 0.72$, $SS(B, B') = 0.85$ and $SS(C, C') = 0.66$, and the costs of transformation of target activities are $C(start) = C(end) = C(A') = 0$, $C(AND - split) = 0.1$, $C(B') = C(C') = 0.2$, $C(D') = 0.4$, $C(AND - join) = 0.1$. In a similar way, the structural similarity degree between q_1 and t_1 is obtained as $\delta_{\gamma_2}(q_1, t_1) = 0.66$ and $\delta_{\gamma_3}(q_1, t_1) = 0.75$. Now, $SS(q, t, M, E) = \min(\delta_{\gamma_2}, \delta_{\gamma_3}) = 0.66$, which means that at least 66% of query activities are mapped to at least a degree 0.66 and at most 66% of target activities have transformation cost to at most 0.66.

Ranking. Consider the preference satisfiability and structural similarity degrees of each potential target presented in Table II. Table III summarizes the results of the different ranking methods discussed in Section VI (where $\omega_{SS} = 0.75$).

The lexicographic order ensures that the first in the ordered list is that having the best structural similarity and, in case of ties, that having the best preference satisfiability. For example t_3 is better than all the other target graphs because its structural similarity is the greatest value. However, a drawback of this method is that the rank can be too drastic, as for the case of $t_5 : (0.78, 0.21)$ and $t_6 : (0.68, 0.72)$. In a such case, the idea of a weighted average is more suitable since it allows for a compensation. Now, with the weighted average t_6 is better than t_5 but generally it does not provide a clear semantics of the induced order. Finally, the min-combination method relies on the worst satisfiability for each service and does not highlight the structural similarity versus the preference satisfiability. The weighted min-combination can overcome the above limitation.

VIII. CONCLUSION

In this paper, we have proposed an approach for web service selection and ranking. In our approach, the evaluation process takes into account two aspects: (i) structural similarity, and (ii) preference satisfiability. User preferences are modelled with fuzzy predicates. Both preference satisfiability and structural similarity are interpreted thanks to linguistic quantifiers. This makes the matchmaking process more flexible and realistic. Some ranking methods have been proposed as well. We are currently working on a prototype system to evaluate our approach by conducting some experiments.

REFERENCES

[1] M. Klusch, B. Fries, and K. Sycara, "Automated semantic web service discovery with owls-mx," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '06, 2006, pp. 915–922.

[2] R. Dijkman, M. Dumas, and L. García-Bañuelos, "Graph matching algorithms for business process model similarity search," in *Proc. of BPM*, 2009, pp. 48 – 63.

[3] D. Grigori, J. C. Corrales, M. Bouzeghoub, and A. Gater, "Ranking bpel processes for service discovery," *IEEE Transactions on Services Computing*, vol. 3, pp. 178–192, 2010.

[4] I. Şora, G. Lazăr, and S. Lung, "Mapping a fuzzy logic approach for qos-aware service selection on current web service standards," in *ICCC-CONTI*, 2010, pp. 553 – 558.

[5] Y. Zhang, H. Huang, D. Yang, H. Zhang, H.-C. Chao, and Y.-M. Huang, "Bring qos to p2p-based semantic service discovery for the universal network," *Personal Ubiquitous Comp.*, vol. 13, no. 7, pp. 471–477, 2009.

[6] D. Dubois and H. Prade, "Using fuzzy sets in flexible querying: Why and how?" in *Proc. of FQAS*, 1996, pp. 89 – 103.

[7] F. Lemos, A. Gater, D. Grigori, and M. Bouzeghoub, "Adding preferences to semantic process model matchmaking," in *GAOC*, 2011.

[8] V. Hristidis, N. Koudas, and Y. Papakonstantinou, "Prefer: A system for the efficient execution of multi-parametric ranked queries," in *SIGMOD Conference*, 2001, pp. 259–270.

[9] J. Chomicki, "Preference formulas in relational queries," *ACM Transactions on Database Systems*, vol. 28, no. 4, pp. 427–466, 2003.

[10] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[11] D. A. D'Mello, I. Kaur, N. Ram, and A. V.S., "Semantic web service selection based on business offering," in *EMS*, 2008, pp. 476 – 481.

[12] S. Agarwal, S. Lamparter, and R. Studer, "Making Web services tradable: A policy-based approach for specifying preferences on Web service properties," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 1, pp. 11–20, Jan. 2009.

[13] M. Sathya, M. Swarnamugi, P. Dhavachelvan, and G. Sureshkumar, "Evaluation of qos based web- service selection techniques for service composition," *IJSE*, vol. 1, pp. 73–90, 2010.

[14] M. Dumas, L. García-Bañuelos, A. Polyvyanyy, Y. Yang, and L. Zhang, "Aggregate quality of service computation for composite services," in *ICSOC*, ser. Lecture Notes in Computer Science, P. P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds., vol. 6470, 2010, pp. 213–227.

[15] W. Kießling, "Foundations of preferences in database systems," in *VLDB*. VLDB Endowment, 2002, pp. 311 – 322.

[16] Z. Wu and M. S. Palmer, "Verb semantics and lexical selection," in *Proc. of ACL*, 1994, pp. 133 –138.

[17] R. R. Yager, "General multiple-objective decision functions and linguistically quantified statements," *International Journal of Man-Machine Studies*, vol. 21, pp. 389 – 400, 1984.

[18] L. Liétard, "A new definition for linguistic summaries of data," in *IEEE World Congress on Computational Intelligence, Fuzzy - IEEE*, Hong-Kong (China), 2008.