HAL
archives-ouvertes.fr

A Cooperative Answering Approach to Fuzzy Preferences Queries in Service Discovery

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

To cite this version:

Katia Abbaci, Fernando Lemos, Allel Hadjali, Daniela Grigori, Ludovic Lietard, et al.. A Cooperative Answering Approach to Fuzzy Preferences Queries in Service Discovery. Flexible Query Answering Systems, 2011, Belgium. 7022, pp.318-329, 2011. <hal-00656155>

HAL Id: hal-00656155

https://hal.archives-ouvertes.fr/hal-00656155

Submitted on 3 Jan 2012

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

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

A Cooperative Answering Approach to Fuzzy Preferences Queries in Service Discovery

Katia Abbaci[1], Fernando Lemos[2], Allel Hadjali[1], Daniela Grigori[2], Ludovic Liétard[3], Daniel Rocacher[1], and Mokrane Bouzeghoub[2]

[1] IRISA/ENSSAT, Rue de Kérampont BP 80518 Lannion, France
{katia.abbaci,allel.hadjali,daniel.rocacher}@enssat.fr
[2] PRiSM Lab, 45 Av. des États Unis 78000 Versailles, France
{fernando.lemos,daniela.grigori,
mokrane.bouzeghoub}@prism.uvsq.fr
[3] IRISA/IUT, Rue Edouard Branly BP 30219 Lannion, France
ludovic.lietard@univ-rennes1.fr

Abstract. In this paper, we propose a novel approach for service retrieval that takes into account the service behavior (described as 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 (Quality of Service) attributes (such as response time, availability and throughput) are modelled by means of fuzzy sets. To avoid empty results, a flexible evaluation method based on fuzzy linguistic quantifiers (such as *almost all*) is introduced. The retrieved results are easily interpreted by the end users thanks to the clear semantics conveyed by that method. Finally, two families of ranking methods are discussed.

Keywords: Cooperative answering, service retrieval, quality of services, fuzzy preferences, linguistic quantifiers.

1 Introduction

Nowadays, an increasing number of companies and organizations are moving towards a service-oriented and model-driven architectures for offering their services on the Web. Searching a specific service within service repositories becomes a critical issue for the success of these architectures. This issue has recently received much attention and many approaches have been proposed [8,2,5]. Most of these approaches are based on the matchmaking of process inputs/outputs [8], service behavior [2] or ontological knowledge [5]. Unfortunately, these approaches often result in a large number of services offering similar functionalities and behavior. One way to discriminate between such similar services is to consider non-functional requirements such as QoS (Quality of Service) (e.g., response time, throughput, availability and reliability). A recent trend towards quality-aware approaches has been initiated [13,1,18], but remains limited and not satisfactory for generic process model discovery.

On the other hand, several service discovery approaches based on fuzzy set theory have been proposed. For instance, in [11] the authors treat the web service selection

H. Christiansen et al. (Eds.): FQAS 2011, LNAI 7022, pp. 318–329, 2011.
© Springer-Verlag Berlin Heidelberg 2011

for composition as a fuzzy constraint satisfiability problem. They assign to each QoS criterion five fuzzy sets (such as *poorly acceptable*, *almost acceptable* and *acceptable*) describing its constraint levels. In [15], QoS based service selection is modelled as a fuzzy multiple criteria decision making problem. Linguistic expressions are used to evaluate and to express the weights of importance of QoS criteria. Hafeez et al. [6] present a service selection mechanism allowing the service broker to intelligently select a set of available services from a user query with imprecise constraints defined by fuzzy sets. The query evaluation is based on the aggregation of the obtained degrees over constraints. Şora et al. [1] propose an approach in which they automatically generate fuzzy rules from user preferences and rank the candidate services using a fuzzy inference process.

The above fuzzy approaches only consider the preference satisfiability and ignore the structural similarity of complex web services. Moreover, these works deal only with services as black boxes, i.e., the service behavior level is not investigated. Our goal is to go further these approaches into a unique integrated approach dealing with functional and non-functional requirements and behavior specification in service retrieval.

Starting from the work done in [9], we propose a cooperative approach for handling users process queries where both behavior specification and QoS preferences are specified inside these queries. User preferences on QoS properties are modelled by means of fuzzy sets as they are more suitable to the interpretation of linguistic terms (such as high or fast) that constitutes a convenient way for users to express their preferences. To avoid empty answers for a given query, a flexible evaluation strategy based on fuzzy linguistic quantifiers is introduced.

The remainder of this paper is organized as follows. Section 2 provides some basic background. In Section 3, modelling fuzzy preferences and their evaluation are addressed. Section 4 presents our interpretation of process models similarity based on linguistic quantifiers. In Section 5, service ranking methods are discussed. Section 6 proposes an illustrative example and finally Section 7 concludes the paper.

2 Background

In this section, we provide some basic definitions within the scope of web service selection with preferences, and a short recall on fuzzy sets.

2.1 Fuzzy Sets

A fuzzy set F [17] 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. The set $\{x \in F | \mu_F(x) > 0\}$ (resp. $\{x \in F | \mu_F(x) = 1\}$) represents the **support** (resp. **core**) of F. In practice, the membership function associated to F is often represented by a trapezoid $(\alpha, \beta, \varphi, \psi)$[1], where $[\alpha, \psi]$ (resp. $[\beta, \varphi]$) is its support (resp. core).

A Fuzzy set-based approach to preferences queries proposed in [3] relies on the use of fuzzy set membership functions that describe the preference profiles of the user on each attribute domain involved in the query. This is especially convenient and suitable

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

when dealing with numerical domains, where a continuum of values is to be interfaced for each domain with satisfiability degrees in the unit interval scale. Then individual satisfiability degrees associated with elementary conditions are combined (commensurability assumption holds thanks to the membership functions) using a panoply of fuzzy set connectives, which may go beyond conjunctive and disjunctive aggregations (by possibly involving fuzzy quantifiers, if only the satisfiability of the most of the elementary conditions in a query is required).

2.2 Preferences in Process Model Specification

Many languages are currently available to describe service process models, e.g., OWL-S [12]. They represent a process model as a set of primitive activities combined using control flow structures. Then, these languages can be abstracted as a direct graph $G = (V, E)$, where the vertices represent activities or control flow nodes, while the edges represent the flow of execution. 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 user query annotated with preferences. The example presents a *global preference* indicating user prefers services providing RSA encryption. Some *activity preferences* are also defined for activities A and B involving reliability, response time and cost. Figure 2 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. In what follows, we present the formal definitions of our model.

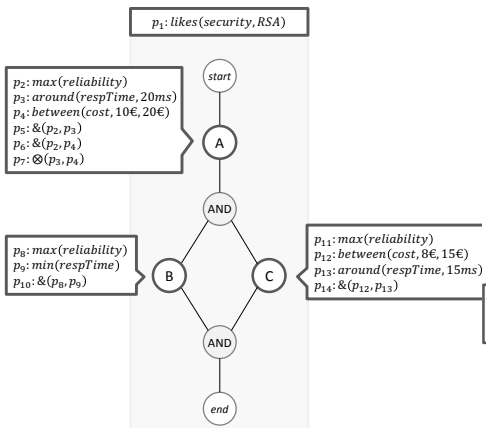


Fig. 1. Query Graph q_1

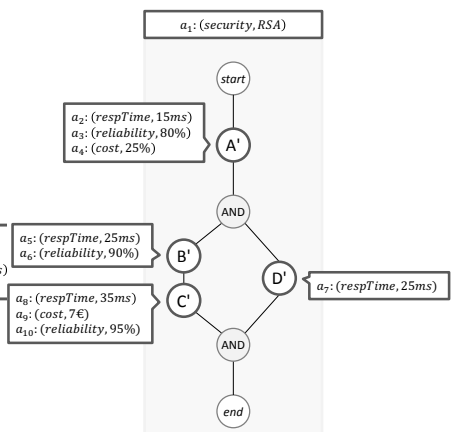


Fig. 2. Target Graph t_1

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**).*

[2] We abstract from the different units in which a value can be described.

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 of the following forms*[3]*:*

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

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

In [9], this set of preferences has been used to develop a service selection approach based on QoS where preference satisfiability is computed using to a unique distance function for all numerical preferences. This way of doing does not take into account the fact that preferences are context and user-dependent and assumes no commensurability when combining individual satisfiability degrees.

3 A Fuzzy Model to Evaluate Preferences

In this section, we introduce a fuzzy set-based approach to handle the above set of preferences involved in the annotated graph associated with the user query. In particular, we propose a metric, called *satisfiability degree* (δ), that measures how well the annotations of a target process model satisfy the preferences present in the query.

[3] Based on a subset of preference operators of the model by [7] that leads to a partial order.

3.1 Atomic Preferences

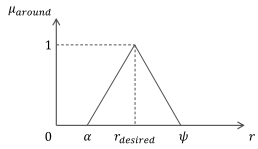
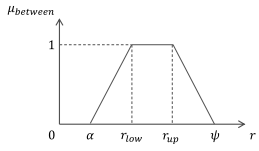
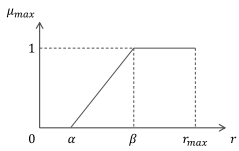
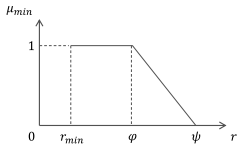
For numerical atomic preferences, the satisfiability degree is obtained using to user-specific membership functions. Table 1 summarizes the fuzzy modelling of numerical preferences of interest. Given a preference p and an annotation $a : (m, r)$, one is interested in computing the degree to which the annotation a satisfies p.

For non-numerical preferences, the satisfiability degree is based on the semantic similarity between concepts. 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 [14]:

$$wp\,(O, c_1, c_2) = \frac{2N_3}{N_1 + N_2 + 2N_3} \tag{1}$$

where c_3 is the least common super-concept between 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 ontology root. Given a non-numerical preference p and an annotation a, the satisfiability degree $\delta\,(p, a)$ is calculated as shown in Table 2.

Table 1. Fuzzy modelling of numerical preferences

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

3.2 Complex Preferences

To evaluate a set of complex preferences S_p, first we construct a *preference tree* t_p that represents the semantics of the set S_p. In that tree, the nodes represent atomic preferences and the edges represent a *more important than* relation (*prioritized preference*)

Table 2. Satisfiability degree of a non-numerical preference p

Non-numerical Preference p	*satisfiability Degree* $\delta(p,a)$
$likes\,(m, r_{desired})$	$\delta\,(p,a) = \begin{cases} 1, & r_{desired} = r \\ wp(O, r_{desired}, r), & otherwise \end{cases}$
$dislikes\,(m, r_{undesired})$	$\delta\,(p,a) = 1 - likes\,(m, r_{undesired})$

from parent to child. Preferences of the same level and having the same parent express *Pareto preference*. Each level i (except the root) of the tree is associated with an importance weight $\omega_i = {}^1/_i$, except $i = 0$ (the smaller i, the more important p_i).

For example, consider the preference tree of q_1 in Figure 3, obtained from the complex preferences of query q_1. 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.

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 [3]. δ'_i is defined [4] using the formula (2).

$$\delta'_i = \max\,(\delta_i, 1 - \omega_i) \tag{2}$$

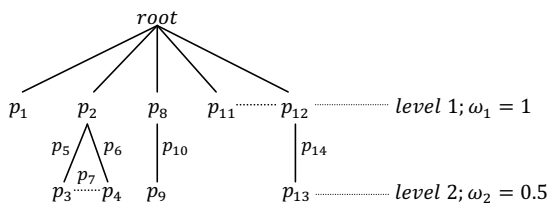


Fig. 3. Sample preference tree

This new interpretation of p_i considers as acceptable any value outside of its support with the degree $1 - \omega_i$. It means 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 calculate the satisfiability degree of user atomic preferences considering their constructors and the complex preferences containing them.

4 Process Model Similarity: A Linguistic Quantifier-Based Method

In this section, we describe a method to compute similarity between process model graphs according to user preferences. We also discuss a method to assess the structural

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

similarity between two process model graphs. Both kinds of similarity will be used to rank potential targets of a query as it will be seen in Section 5.

In order to evaluate the structural similarity of two processes q and t, we propose to use a graph matching algorithm, like in [5]. This algorithm returns a mapping M and a set E of edit operations necessary to transform q into t. We consider a mapping M between q and t as a set of pairs (v, w), such that v is an activity of q and w is an activity of t or the symbol \$, which indicates the deletion of v. The edit operations considered are simple graph edit operations: node/edge deletion, node/edge addition and node substitution. Figure 4 illustrates a mapping between a query graph q_1 and a target graph t_1. In the figure, $SS\,(v, w)$ denotes the semantic similarity between activities v and w; we use the metric proposed in [5] that considers the activity name, inputs and outputs. In our work, the preference evaluation explained in Section 3 is applied as follows: the global atomic preferences of q are evaluated against the global annotations of t; similarly, the atomic preferences of an activity v of q are evaluated against the annotations of an activity w of t, such that $(v, w) \in M$.

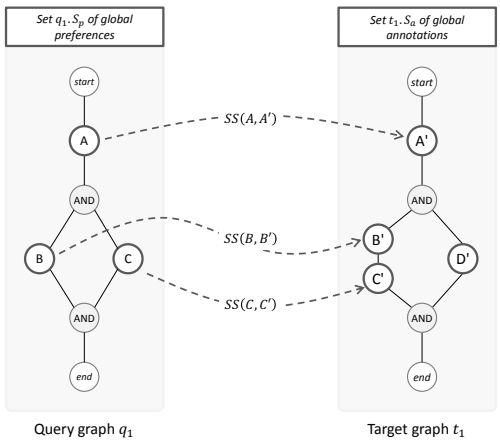


Fig. 4. Sample mapping M between a query graph q_1 and a target graph t_1

In our approach, we rely on the linguistic quantifier *"almost all"* for the similarity evaluation process. Such a quantifier, which is seen as a relaxation form of the universal quantifier "*all*", constitutes an appropriate tool to avoid the empty answers. It allows to retrieve elements that would not be selected when the quantifier "*all*" is used.

4.1 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:

$$\gamma_1\text{: Almost all preferences of } q \text{ are satisfied by } t$$

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.) [4]; (ii) X is a set of elements; (iii) P is a fuzzy predicate. The truth degree δ_γ of $\gamma :\ Q\ X\ are\ P$ is computed according to Yager's method [16]:

- Let $\Omega = \{\mu_1, \ldots, \mu_n\}$ be a set of degrees of the elements of X w.r.t. P, ordered in decreasing way: $(\mu_1 \geq \ldots \geq \mu_n)$.
- The truth degree δ_γ is given by formula (3), where $\mu_Q\ (i/n)$ is a membership degree of the element i/n to Q.

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

In our case, $\Omega = \{\mu_1 : \delta'_1, \ldots, \mu_n : \delta'_n\}$ is the set of satisfiability degrees of all atomic preferences (i.e. all global and activity atomic preferences) of query q, where δ'_i is the satisfiability degree of an atomic preference p_i computed by formula (2). The semantics of the linguistic quantifier *almost all* is given by (50%, 80%, 100%, 100%). In this case, (i) the user is totally satisfied if at least 80% of preferences are satisfied and (ii) the user is not satisfied at all if at most 50% of preferences are satisfied.

4.2 Structural Similarity between Process Models

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

$$\begin{cases} \gamma_2 : \text{ Almost all the activities of } q \text{ are mapped with activities of } t \\ \gamma_3 : \text{ Almost no edit operation is necessary to transform } q \text{ into } t \end{cases}$$

The truth degree of proposition γ_2 is obtained from the formula (3), 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 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:

$$\delta_{\gamma_3} = \max_{1 \leq i \leq n} \min\left(1 - \mu_i, 1 - \mu_Q\ (i/n)\right) \tag{4}$$

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.

Thus, the structural similarity between q and t is evaluated as follows:

$$SS = \min\left(\delta_{\gamma_2}, \delta_{\gamma_3}\right) \tag{5}$$

Remark. In our approach, we consider particularly formulae (3) and (4) where $\mu_Q\ (i/n) = i/n$. Thus, the meaning of delivered degrees has a simple and clear semantics for the user [10]. 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 δ_{γ_1}, at least $\delta^*_{\gamma_2}$% of activities of q are mapped with t to at least a degree δ_{γ_2} and at least $\delta^*_{\gamma_3}$% of q does not need edit operation to transform q into t to at least a degree δ_{γ_3} (where $\delta^*_{\gamma_i} = 100 \times \delta_{\gamma_i}$).

5 Process Model Ranking

In this section, given a set of target graphs that are relevant to a query, we discuss some methods to rank-order these graphs according to their structural similarity and preference satisfiability. Let $\delta\,(q, t, M)$ be the satisfiability degree between query graph q and target graph t obtained by formula (3) according to a mapping M. Similarly, let $SS\,(q, t, M, E)$ be the structural similarity between q and t obtained by formula (5) w.r.t. M and a set E of edit operations. Two kinds of ranking methods can be used.

Ranking methods based on aggregation. In this first category, ranking methods aggregate both structural similarity and preference satisfiability into a unique degree used to rank-order the target graphs. Two aggregations can be considered:

Weighted average-based aggregation. The weighted average of $SS\,(q, t, M, E)$ and $\delta\,(q, t, M)$ is given by equation (6).

$$rank\,(q, t) = \omega_{SS} \times SS\,(q, t, M, E) + (1 - \omega_{SS}) \times \delta\,(q, t, M) \tag{6}$$

s. t. $0 < \omega_{SS} < 1$ is an importance weight assigned to the structural similarity criterion.

Min-combination based aggregation. The min-combination method [17] selects the smallest value of the two similarity degrees $SS\,(q, t, M, E)$ *and* $\delta\,(q, t, M)$, i.e.,

$$rank\,(q, t) = \min\,(SS\,(q, t, M, E)\,, \delta\,(q, t, M)) \tag{7}$$

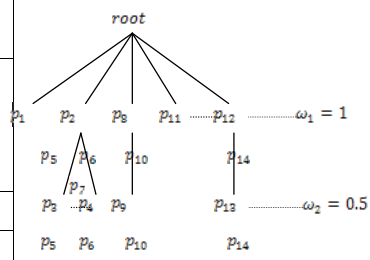
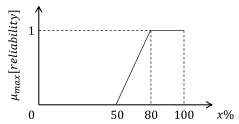
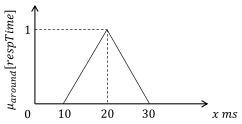
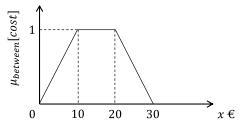
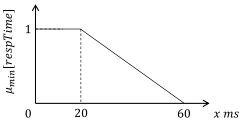
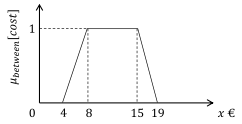
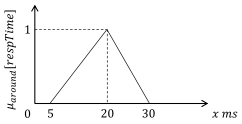
Ranking method without aggregation. In this second category, the two distinct similarity degrees are used to rank-order target graphs thanks to the *lexicographic order*. A priority is given to the structural similarity $SS(q, t, M, E)$ while the preference satisfiability $\delta\,(q, t, M)$ is only used to break ties.

6 Illustrative Example

We give here an example of service discovery for query q_1 of Figure 1. First, we compute the preference satisfiability degree between q_1 and the potential target graphs. To illustrate, we evaluate the preference satisfiability degree between q_1 and target t_1 of Figure 2. We consider the mapping between them as depicted in Figure 4. Then, we apply the ranking methods described in Section 5. See below for more details.

First, we compute the satisfiability degree δ of user preferences as shown in Table 3. Consider, for example, the pair (A, A') in Table 3. The satisfiability degree $\delta\,(p_2, a_2)$ between preference p_2 and annotation a_2 is obtained by $\mu_{max}\,[reliability]$. According to equation (2), a preference tree allows to aggregate the preference degrees of A. The result is presented in column RESULT in Table 3. Second, we apply the truth degree

Table 3. Satisfiability degrees of each pair of matched activities

SATISFIABILITY DEGREE CALCULATION				
ATOMIC PREFERENCES			COMPLEX PREFERENCES	
PREF.	MEMBERSHIP FUNCTION	δ_i	PREFERENCE TREE	δ'_i
p_1		$\delta(p_1, a_1) = 1$		$\delta'_1 = 1$
p_2		$\delta(p_2, a_3) = 1$		$\delta'_2 = 1$
p_3		$\delta(p_3, a_2) = 0.5$		$\delta'_3 = 0.5$
p_4		$\delta(p_4, a_4) = 0.5$		$\delta'_4 = 0.5$
p_8	The same of preference p_2.	$\delta(p_8, a_6) = 1$		$\delta'_8 = 1$
p_9		$\delta(p_9, a_5) = 0.9$		$\delta'_9 = 0.9$
p_{11}	The same of preference p_2.	$\delta(p_{11}, a_{10}) = 1$		$\delta'_{11} = 1$
p_{12}		$\delta(p_{12}, a_9) = 0.75$		δ'_{12} =0.75
p_{13}		$\delta(p_{13}, a_8) = 0$		$\delta'_{13} = 0.5$

described in Section 4.1 to obtain the global satisfiability degree between q_1 and t_1, as follows: $\delta_{\gamma_1}(q_1, t_1)$ = max (min $(1, \mu_Q(1/9))$, ..., min $(0.5, \mu_Q(9/9))$) = 0.67. This means that at least 67% of preferences are satisfied to at least a degree 0.67.

Assume now that the semantic 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 $C(start) = 0$, $C(A') = 0$, $C(AND - split) = 0.1$, $C(B') = 0.2$, $C(C') = 0.2$, $C(D') = 0.4$, $C(AND - join) = 0.1$ and $C(end) = 0$. In a similar way, the structural similarity degree between q_1 and t_1 is obtained as: $\delta_{\gamma_2}(q_1, t_1)$ = max (min $(0.85, \mu_Q(1/3))$, ..., min $(0.66, \mu_Q(3/3))$) = 0.66 and $\delta_{\gamma_3}(q_1, t_1)$ = max (min $(1 - 0.4, 1 - \mu_Q(1/8))$, ..., min $(1 - 0, 1 - \mu_Q(8/8))$) = 0.75.

Table 4. Structural similarity and preference satisfiability degrees of a set of target graphs

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	0.35
t_5	0.78	0.21
t_6	0.68	0.72
t_7	0.66	0.72
t_8	0.66	0.35

Table 5. Ranking of target graphs according to the three ranking methods

WEIGHTED AVERAGE		MIN-COMBINATION		LEXICOGRAPHIC ORDER
t_3	$wa = 0.74$	t_6	$mc = 0.68$	t_3
t_6	$wa = 0.69$	t_7	$mc = 0.66$	t_4
t_7	$wa = 0.68$	t_1	$mc = 0.66$	t_5
t_4	$wa = 0.67$	t_3	$mc = 0.40$	t_6
t_1	$wa = 0.66$	t_4	$mc = 0.35$	t_7
t_5	$wa = 0.64$	t_8	$mc = 0.35$	t_8
t_8	$wa = 0.58$	t_2	$mc = 0.29$	t_1
t_2	$wa = 0.40$	t_5	$mc = 0.21$	t_2

Now, $SS\left(q, t, M, E\right) = \min\left(\delta_{\gamma_2}\left(q_1, t_1\right), \delta_{\gamma_3}\left(q_1, t_1\right)\right) = 0.66$. It 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.

As presented in Table 4, eight potential answers to query q_1 are retrieved. Table 5 summarizes the results of the different ranking methods discussed in Section 5 (where $\omega_{SS} = 0.75$).

7 Conclusion

In this paper, we have proposed an approach for web services selection and ranking where both structural similarity and preference satisfiability are taken into account in the evaluation step. User preferences are modelled thanks to fuzzy predicates while linguistic quantifiers are used as a basis to compute the process model similarity. So, the matchmaking process is achieved in a more cooperative and flexible way. Some ranking methods have been discussed in the scope of services retrieval. We are currently working on a prototype system to evaluate our approach by conducting some experiments.

References

1. Şora, I., Lazăr, G., Lung, S.: Mapping a fuzzy logic approach for qos-aware service selection on current web service standards. In: ICCC-CONTI, pp. 553–558 (2010)
2. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)
3. Dubois, D., Prade, H.: Using fuzzy sets in flexible querying: Why and how? In: Proc. of FQAS, pp. 89–103 (1996)
4. Glöckner, I.: Fuzzy Quantifiers in Natural Language: Semantics and Computational Models. Der Andere Verlag, Osnabrück (2004)
5. Grigori, D., Corrales, J.C., Bouzeghoub, M., Gater, A.: Ranking bpel processes for service discovery. IEEE Transactions on Services Computing 3, 178–192 (2010)
6. Hafeez, O., Chung, S., Cock, M.D., Davalos, S.: Towards an intelligent service broker with imprecise constraints: Fuzzy logic based service selection by using sawsdl. Tcss 702 design project in computing and software systems. University of Washington (2008)

7. Kießling, W.: Foundations of preferences in database systems. In: VLDB, pp. 311–322. VLDB Endowment (2002)
8. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with owls-mx. In: Proc. of AAMAS, pp. 915–922 (2006)
9. Lemos, F., Gater, A., Grigori, D., Bouzeghoub, M.: Adding preferences to semantic process model matchmaking. In: Proc. of GAOC (2011)
10. Liétard, L.: A new definition for linguistic summaries of data. IEEE World Congress on Computational Intelligence, Fuzzy-IEEE, Hong-Kong, China (2008)
11. Lin, M., Xie, J., Guo, H., Wang, H.: Solving qos-driven web service dynamic composition as fuzzy constraint satisfaction. Proc. of EEE, 9–14 (2005)
12. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T.R., Sirin, E., Srinivasan, N., Sycara, K.: Owl-s: Semantic markup for web services
13. Mokhtar, S.B., Preuveneers, D., Georgantas, N., Issarny, V., Berbers, Y.: Easy: Efficient semantic service discovery in pervasive computing environments with qos and context support. Journal of Systems and Software 81(5), 785–808 (2008)
14. Wu, Z., Palmer, M.S.: Verb semantics and lexical selection. In: Proc. of ACL, pp. 133–138 (1994)
15. Xiong, P., Fanin, Y.: Qos-aware web service selection by a synthetic weight. In: Proc. of FSKD, vol. (3), pp. 632–637 (2007)
16. Yager, R.R.: General multiple-objective decision functions and linguistically quantified statements. International Journal of Man-Machine Studies 21, 389–400 (1984)
17. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
18. Zhang, Y., Huang, H., Yang, D., Zhang, H., Chao, H.-C., Huang, Y.-M.: Bring qos to p2p-based semantic service discovery for the universal network. Personal Ubiquitous Computing 13(7), 471–477 (2009)