QoS-based Approach for Context-aware Service Selection

Mohcine Madkour, Driss Elghanami, Abdelilah Maach, Hasbi abderrahim
Department of Computer Science
Mohamadia School of engineers at Mohamed 5 University
BP:765 Avenue Ibn Sina
Rabat, Morocco

ABSTRACT: *In ubiquitous computing the discovery process may retrieve many services when in fact only one of them fit exactly user satisfaction, besides, after discovering is achieved, there is a set of candidate services between theme a selection must be made. In fact, discovery is a prerequisite for selection, but selection is the main purpose. Actually, uncertainty of context information may lead to inexact matching between already discovered and required service capabilities, and consequently to the non selection of fitting services. In order to handle incomplete context information, we propose in this paper a workflow-based algorithm allowing inexact matches for matching contextual service descriptions using similarity measures. Service description and request are compared using four kinds of similarity measures: syntactic, linguistic, structural and QoS semantic measures, which compare individually requested and provided properties represented as workflow nodes, and thereafter the global measures which take into account context and service as a whole are aggregated by means of the linguistic quantifier "almost all". In our approach we consider that functional aspects of a service are already met and we focus on non-functional and QoS-related aspects of service description to rank-order the discovered services.*

Keywords: Non functional context, Context-awareness, Service selection, Service discovery, Workflow Graphs, Similarity Measures, Linguistic Quatifier

Received: 29 June 2012, Revised 8 August 2012, Accepted 12 August 2012

1. Introduction

The term Ubiquitous Computing, introduced by Weiser [1], refers to the seamless integration of devices into users' everyday life. This term represents an emerging trend towards environments composed of numerous computing devices typically mobile or embedded and connected to a network infrastructure composed of a wired core and wireless edges. In pervasive scenarios perceived by Ubiquitous Computing, searching for a specific service within service repositories becomes 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 and service behavior (described as a process model). Nevertheless, these approaches still remain with a high selection rate, ensuing in a huge number of services presenting similar applications and behavior. One way to reduce the selection rate is to cope with non-functional context such as quality attributes.

More specifically, in ubiquitous environments, context information is naturally uncertain and fuzzy and hence service matchmaking mechanisms have to cope with the fuzziness issue by dealing with questions such how to model the inaccuracy of the context values, how to handle fuzzy predicates of context in service matching and how to aggregate calculated atomic similarities measures values.

In this paper we propose a QoS-based approach for context-aware service selection, our approach take into account non-functional context attributes. In view of that the non-functional context is used for ranking of the provided services which are analyzed as graphs whose objects represent concepts and properties and edges represent the relations connecting such concepts.

In the workflow based representation potential services act like a "*targets*" while the execution context environment is the request for the service, afterwards the two graphs are compared in term of the syntactic, structural, linguistic and QoS semantic similarities such that in the ranking mechanism we rank-order the suited services based on the similarities criteria.

In the remainder of this paper, Section 2 discusses the role of QoS attributes in service selection and proposes a workflow based model for context and service processes models. Section 3 presents the four measures used for comparing the similarities between the processes models. Section 4 addresses our linguistic quantifier based method for aggregating similarities measures. Section 5 shows a discussion about process models ranking and Section 6 concludes this paper.

2. Behavioral and QoS semantic approach in service selection

2.1 Beyond service behavior: Quality attributes

Service descriptions are used to advertise the service capabilities, interface, behavior, and quality. Publication of such information about available services provides the necessary means for discovery and selection of services. In particular, the service capability description states the conceptual purpose and expected results of the service (by using terms or concepts defined in an application-specific taxonomy). The service interface description publishes the service signature (its input/output/error parameters and message types). The (expected) behavior of a service during its execution is described by its service behavior description (for example, as a workflow process model). Finally, the Quality of Service (QoS) description publishes important functional and nonfunctional service quality attributes, such as service metering and cost, performance metrics (response time, for instance), security attributes, (transactional) integrity, reliability, scalability, and availability.

On the other hand, in the presence of multiple services with overlapping or identical functionality, service requesters need objective QoS criteria to distinguish one service from another. We argue that it is not practical to come up with a standard QoS model that can be used for all web services in all domains. This is because QoS is a broad concept that can encompass a number of context-dependent non-functional properties such as privacy, reputation and usability. Moreover, when evaluating QoS of web services, we should also take into consideration domain specific criteria.

Moreover, quality of service [2] plays an important role in automatic web service selection. It is mainly used to establish valid and reliable web service and identity the best offers systematically from a set of functionally similar ones. QoS parameter gives user assurance and confidence to use the service.

There are many research efforts to define and categorize QoS as well as attempts to express, quantify, and model QoSs [3,4,5,6,7].Garcia and de Toledo [8] define a set of major Web Service QoS attributes. Menasce [9] examines QoS issues in Web Services, and Yu et al. [10] provide a list of QoS parameters and explain how to evaluate each. Although security is not a quantifiable QoS, these authors present a formula to test the security of Web Services based on the number of attack detections. Based on the most common QoS requirements in the literature, Rahman and Meziane[11] present five essential QoS requirements: readiness, transaction, reliability, speed, and security. Table 1 presents the description of some QoS parameters used in this work.

2.2 A Model of Semantic QoS Specifications

Service retrieval based on functional semantic attributes is not satisfactory for a great number of applications. Our Retrieval proposals address both the capabilities and behavioral service description levels. Yet, taking also into account the QoS semantic level is mandatory to enable the fully-automatic retrieval of services from service repositories. We propose a new selection technique that, compared to related work, supports both behavioral and semantic QoS service description levels, works system-wide, and yields a good retrieval rate.

QoS parameters	Descriptions
Latency	"*The time the SOAP message needs to reach its destination*".
Execution Time	The time taken by the service to execute and process its sequence of activities.
Response Time	The time required to process and completes a service request; the response time includes the execution time and the latency.
Throughput	The number of requests a service can process per unit of time.
Security	Offers mechanisms of authentication, authorization, confidentiality, non-repudiation, accountability, traceability, and audit ability.

Table 1. Used QoS parameters and descriptions

To give a detailed perspective on how to interact with a service, it can be viewed as a process. Self [12] showed that focusing on the process by which knowledge is constructed is more important than focusing on the target knowledge itself. Specifically, specifying context-aware service as process models draw upon well-established work in a variety of fields, including work in AI on standardizations of planning languages, work in programming languages and distributed systems, emerging standards in process modeling and workflow technology such as the NIST's Process Specification Language (PSL) [13] and the Workflow Management Coalition effort[1]work on modeling verb semantics and event structure [14], work in AI on modeling complex actions [15], and work in agent communication languages [16].

Process can be categorized as "*Atomic*", "*Composite*", or "*Simple*". A simple procedure can be represented as a single atomic process, while a complicated procedure can be represented as a composite process (or several composite processes). The latter can be further decomposed into many composite processes or atomic processes. The advantage of this model is that it presents different views of the same procedure (i.e. either a higher view or a detailed view of the same procedure).

The key to effective retrieval precision is capturing enough service and query semantics to substantively increase precision without making it unrealistically difficult for people to express these semantics. Our central claim is that we can achieve these goals using process models. A process model captures behavior as a collection of interlinked subactivities. So To understand why we use process models, we need to understand the causes of imperfect precision between query and service process models formulation such as: An imperfect precision can occur when a part of a query or service description is taken to have an unintended role. For example, a keyword-based query to find Renting car services would also match descriptions such as "*easy car booking*". An imperfect precision can also occur when two or more parts of a query or service model are taken to have an unintended relationship. Like do not distinguishing between rent car and buy car.

In this work, we define four similarities types to allow an approximate matching between process models even with imprecise in texts formulation or in semantic logic interpretation or also in structural form differentiation.

2.3 Workflow-based process model with QoS specifications for service selection

Different formal models representing process models exist, such as Workflow Nets [17], Finite State Automata [18], Petri Nets [19], or stat-echarts [20]. The most of service behavioral models are workflow-based view their semantic richness and their base discovery algorithms propose heuristics to reduce execution time. The workflow model is an extension of Finite State Automata and it has been introduced as annotated Finite State Automata, it allows capturing many control flow structures (e.g., parallelism) and generally it doesn't blow up with small-sized graphs.

In the figure 11 bellow an example of automata and workflow based service process models

We define a workflow as a graph, where the vertices represent "*activities*" and the edges represent the precedence between the "*activities*". The graph may be cyclic and the graph needs not be connected. Associated with vertex, edge or subgraph of the graph, annotations describe the operations performed on the workflow, provide the provenance of the data computed by the workflow and specify the nature of the dependency between the vertices. The following figure12 and figure 13 shows an example of two workflows for service request and service applicant respectively represented as graphs annotated with Semantic QoS attributes.

[1] http://www.aiim.org/wfmc

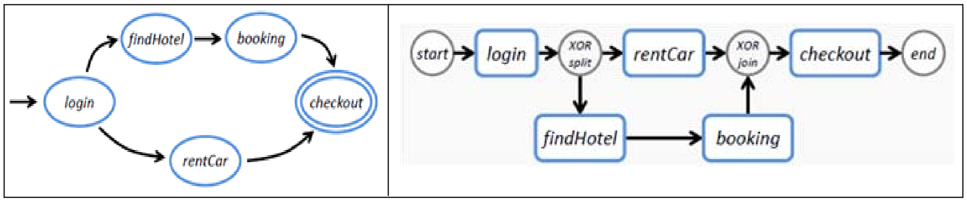


Figure 1. Examples of automate and workflow graphs

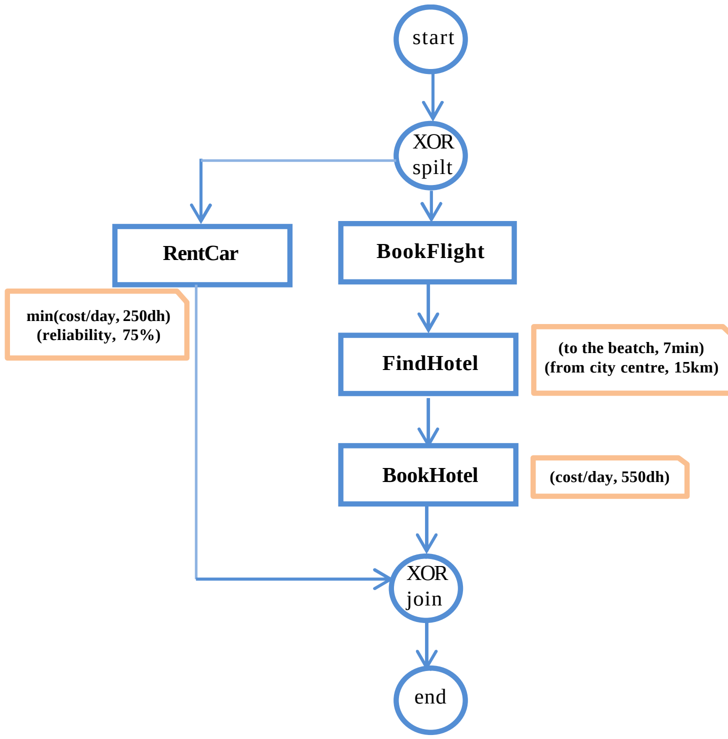


Figure 2. Request workflow graph *r*

We point out that, in this work, service process models are considered already annotated with QoS attributes while the context manager is the part that defines the semantic QoS annotations of its request. We do not argue here the techniques to obtain the QoS information of a process model. For this, look at the work in [21].

In this context, similarity search is deûned as follows: given a process model q (the query) and a collection of process models targets T, retrieve the models in T that are most similar to p and rank them according to their degree of similarity. Similarity searches are deûned with respect to similarities measures between query and current service process models. The similarity between pairs of process models can be measured on the basis of four complementary aspects of process models: (i) (ii) the syntax and language of labels attached to tasks, events and other model elements; (iii) their graph structure; (iiii) their semantics QoS satisfaction. The next section discusses the similarities computing techniques.

3. Similarity measures computing

In order to efficiently select service based on their process models from model repository we should develop an efficient similarity measure based systems, we theoretically define four similarities measures between process models: syntactic similarity, linguistic similarity, and structural similarity and QoS semantic similarity

Similar process models have similar behavior; the similarity might represent an indication on how convenient a service process model will be, when it is applied to answer the context query. Similarity measures as explained in this paper is a real-valued

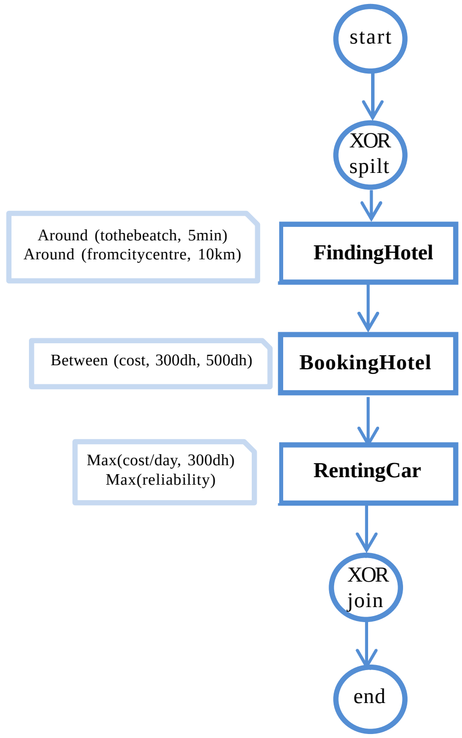
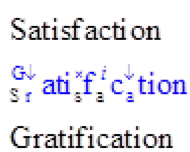


Figure 3. Service applicant workflow graph

function sym: $S \times S \rightarrow [0, 1]$ on a set S measuring the degree of similarity between two elements.

3.1 Syntactic Similarity Measure

In order to measure similarity between the labels of two nodes in the workflows, Levenshtein [22] proposed the string-edit distance method to the measure of syntactic similarity. The string-edit distance is the number of atomic string operations necessary to get from one string to another. These atomic string operations include: removing a character, inserting a character or substituting a character for another. For example, the edit distance between the strings "*Satisfaction*" and "*Gratification*" equals 5, because five substitutions are sufficient to transform "*Satisfaction*" to "*Gratification*".



The greater the edit distance, the more different the strings are. Based on this edit distance(ed) method, R. Dijkman& M.Dumas [23] have proposed a syntactic similarity measure as shown below, which returns similarity degrees between 0 and 1, where 1 stands for perfect match and zero for bad match.

Let n_1 and n_2 be two nodes from those workflows and let $l_1(n_1)$, $l_2(n_2)$ be the two strings that represent the labels of those nodes, i.e. we can calculate their length, denoted $| l_1(n_1) |$, $| l_2(n_2) |$ and their edit distance denoted $ed\ (l_1(n_1), l_2(n_2))$.We define the syntactic similarity of workflow nodes n_1 and n_2 as follows:

$$\text{SYN-Sym}\,(\text{n}_1, \text{n}_2) = 1\ \frac{ed\ (l_1(n_1), l_2(n_2))}{\max(|\,l_1(n_1)\,|, |\,l_2(n_2)\,|)}$$

This measure considers the number of shifts being made to change one string into the other and weights the number of these changes against the length of the highest string of these two strings $\max(|\,l_1(n_1)\,|, |\,l_2(n_2)\,|)$.

3.2 Linguistic Similarity Measure

Linguistic similarity is a confidence score that reflects linguistic relation between the meanings of two sentences. To exploit linguistic features we have utilized WordNet [24]. WordNet is a machine-readable lexical database which is organized by meanings and developed at Princeton University. The words in Wordnet are classiûed into four categories, nouns, verbs, adjectives and adverbs respectively. WordNet groups these words into sets of synonyms called synsets, provides short definitions, and records the various semantic relations between these synonym sets. Synsets are interlinked by means of conceptual and lexical relations such hyperonyms (generalization) andhyponyms (specialization). Figure 14 is part of the hierarchy of WordNet.

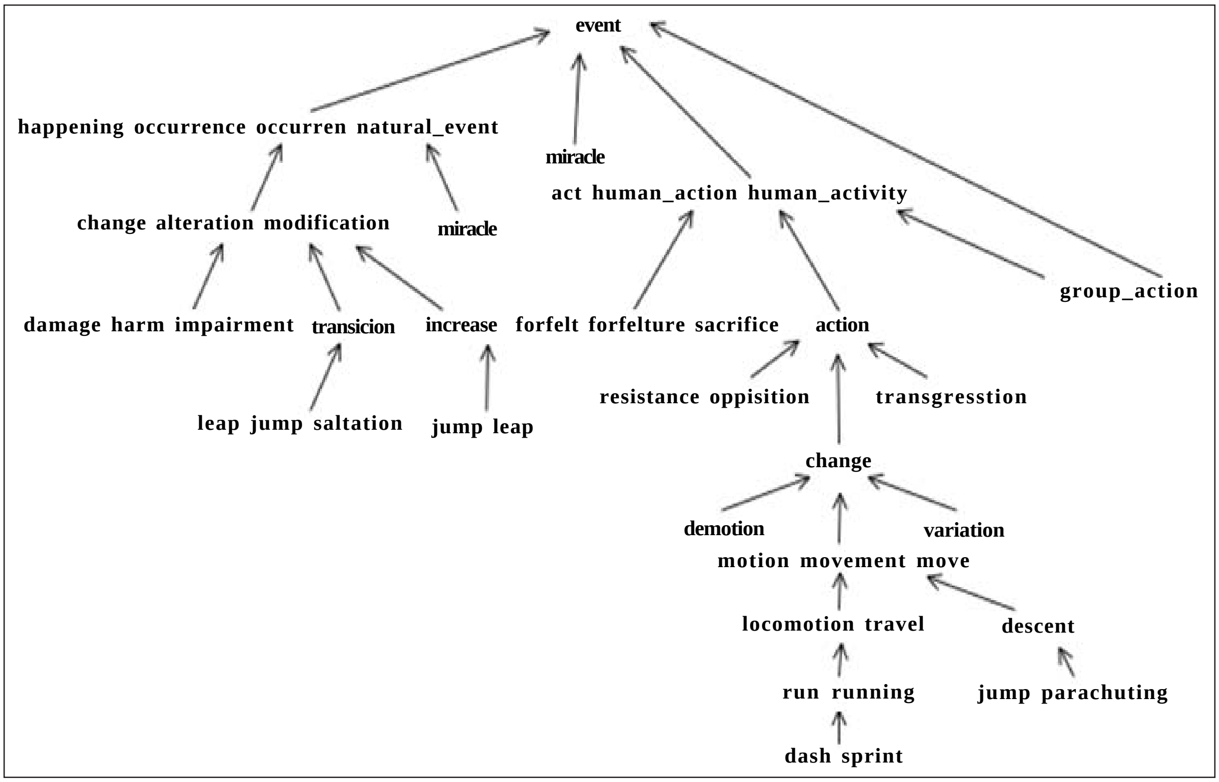


Figure 4. Example part of wordnet

To compute linguistic similarity degrees between process element names we are considering synonym sets proposed by WordNet. For instance, WordNet provides for the term verification two senses in a synonym relationship ordered by estimated frequency:

• **Satisfaction:** satisfaction # (the contentment one feels when one has fulfilled a desire, need, or expectation) satisfaction #2: Gratification#1 (state of being gratified or satisfied) satisfaction #3: atonement expiation (compensation for a wrong) satisfaction #4: ((law) the payment of a debt or fulfillment of an obligation) satisfaction# 5 (act of fulfilling a desire or need or appetite).

• **Gratification:** gratification#1: satisfaction#2 (state of being gratified or satisfied) gratification#2 (the act or an instance of satisfying).

One synonym for satisfaction is gratification (sense 2). Thus, our implemented system indicates some linguistic relationships between these two instance names. Modeling business processes depends on the modeler and varies from one to another. One modeler might denote elements only with verbs while someone else might denominate elements with labels composed of nouns, verbs and adverbs (e.g. confirm the satisfaction). In case of composed process element names we calculate only the linguistic similarity for names that do not satisfy a syntactic similarity of 1.0.For instance, computing linguistic similarity for the pair (confirm satisfaction vs. confirm gratification) is restricted to the pair satisfaction vs. gratification due to identity of send, generally in case where there is phrases of more than word in the nodes we compare normally potentials words in this phrases, potential doesn't come always with the meaning of different but the word which handle the main meaning of the phrase. In case when there is more than one potential word in the node we make the global score of linguistic similarity as the average of all the potential pair linguistic similarity. In the following we suppose that we have only one potential word in two phrases labels.

Let l_1 and l_2 be the labels respectively of two nodes n_1 and n_2 from the workflows and let $w_1 = \omega(l_1)$ and $w_2 = \omega(l_2)$ be the retrieved sets of senses of l_1 and l_2 and $|w_1|$ and $|w_2|$ are their number of words, the function ω retrieves all senses of a given word from WorldNet. Let $S = w_1 \cap w_2$ be the set of common senses of the phrasings of n_1 and n_2. We have

$$f(S) = \begin{cases} 1 \text{ if } w_1 \cap w_2 \neq \varnothing \\ 0 \text{ if } w_1 \cap w_2 \neq \varnothing \end{cases}$$

Then we define the linguistic similarity between the nodes and as:

$$\text{Lin-Sym}(n_1, n_2) = \frac{f(S)}{\max(|w_1|, |w_2|)}$$

This measure considers synonym relationships in the Wordnet database, for our measure we implicate the synonyms that are proposed for both labels and weight the intersection of synonyms against the maximum sense cardinality of these two labels. This measure returns similarity degree of 1.0 for l_1 vs. l_2 if is the only synonym for l_1 and l_2 vice versa.

3.3 Structural similarity measure

Early works on structural pattern recognition literature was made by Shapiro and Haralick [25] who showed how inexact structural representations could be compared by counting consistent subgraphs, this similarity measure was refined after by Eshera and Fu [26] and by Sanfeliu and Fu [27] who showed how the concept of string edit distance could be extended to graphical structures.

In its structure, the workflow can be modified to fulfill some requirements in structural similarity measurement between graphs relative to the model process query. In general, the modification operation can be one these:

• Add nodes or edges to the graph for example, information about the dependency between two nodes can be described by adding nodes and edges with annotations

• Delete nodes or edges from the graph, for example, activities that have already been executed (based on provenance information) can be deleted from the workflow. For example, the editor can delete redundant works in workflow execution by removing vertices and edges from a workflow, so-called '*Graph Reduce*'.

• Substitute nodes or edges by modifying annotations on the graph, nodes or edges. For instance, data or execution planning editor performs the modification with replacing a logical name with a physical name for data or resource location.

By considering process models as labeled graphs G = (N, E) where N is the nodes and E is the edges, We can then assign a similarity score to two graphs by computing their graph-edit distance [28]. The graph edit distance between two graphs is the minimal number of graph edit operations that is necessary to get from one graph to the other. The different graph edit operations can be considered, for this aim, we need the notion of graphs mapping.

Let $G_1 = (N, E)$ and $G_2 = (N, E)$ be two workflow graphs of process models, the mapping between G_1 and G_2 is defined as the partial injective mapping $M: (N_1 \rightarrow N_2) \cup (E_1 \rightarrow E_2)$ that maps nodes and edges from one graph to another. In this light, to go from the graph G_1 to graph G_2 three operations are allowed: insertion node or edge, deletion node or edge and substitution nodes or edges (the alteration of the label of node or edge), let ID be the set of all inserted and deleted nodes or edges and SB be the set of all substituted nodes or edges, The distance induced by the mapping is defined as $D = |ID| + |SB| + \sum_{n,m \in M} 1 - \text{syn-sym}(n, m)$ with sun-sym means the syntactic similarity .The graph edit distance is the minimal possible distance induced by a mapping between the two processes.

The graph edit distance similarity is computed as one minus the average of the fraction of inserted or deleted nodes, the fraction of inserted or deleted edges and the average distance of substituted nodes. We define the graph edit distance similarity as:

$$\text{STR-Sym} = 1 - \left\{ \overline{\frac{|ID|}{|N_1|+|N_2|}, \frac{|SB|}{|E_1|+|E_2|}, \frac{2\sum_{n,m \in M} 1 - \text{syn-sym}(n,m)}{|N_1|+|N_2| - ID}} \right\}$$

In case of a user is experimented with insertion and deletion variation, he can opt for the weighted average in the place of plain average.

2.4 Semantic QoS Similarity measure

Each service query process model supports a set of QoS requirements for the context situation like throughput, reliability, response time and cost, these QoS attributes are always attached to its semantic values, for instance the "*reliability* " is a vague attribute if it's not set in its semantic status, but we need somewhat to measure these quality attributes in process models query.

Normally the majority of QoS attributes of context are not exactly value-fixed in service request, from our experience in fuzzy querying of services process models, the QoS context attribute can be expressed as:

Atomic expressions:

• *around* (t, $v_{desired}$, μ_{around}) for attribute t, this expression emphasizes the value $v_{desired}$, otherwise, it privileges those close to $v_{desired}$. The membership function μ_{around} evaluates the degree to which a value v satisfies v_{desired},

• *between* (t, v_{low}, v_{up}, $\mu_{between}$): for attribute t, it emphasizes the values inside the interval [t_{low}, t_{upper}], otherwise, it favors the values close to the limits. The function $\mu_{between}$ evaluates the degree to which a value v satisfies the interval [v_{low}, v_{upper}],

• *max* (t, μ_{max}): for attribute t, it emphasizes the highest value; otherwise, the closest value to the maximum is favored. The function μ_{max}: evaluates the degree to which a value r satisfies the highest value of t;

• *min* (t, μ_{min}): for attribute t, it emphasizes the lowest value; otherwise, the closest value to the minimum is favored, μ_{min} evaluates to which degree a value v satisfies the lowest value of t;

• *likes* (t, $v_{desired}$): for attribute t, it emphasizes the value $v_{desired}$; otherwise, any other value is accepted;

• dislikes (t, $v_{undesired}$): for attribute t, it emphasizes the values that are not equal to $v_{undesired}$, otherwise, $v_{undesired}$ is accepted;

Figure 1 is a sample example from scenario; it illustrates the user request annotated with certain quality attributes involving reliability, response time and cost.

Naturally the expression of quality attribute can be categorized as: numerical (around, between, max, min) and non-numerical (likes, dislikes). The values of the non-numerical attributes come often from the context ontology and they often describe the global QoS desire of the user over his tasks.

In the other hand services should be integrated with semantic web by annotating the various semantic descriptions of services. To do so, the applicants process models are all annotated by valuated semantic QoS attributes, these annotation can be over the workflow graph nodes or over subgraphs of the workflow graph, Indeed the represented attributes compose both functional and non-functional QoS properties; functional properties can be measured in terms of throughput, latency, response time; non-functional properties address various issues including integrity, reliability, availability and security of services.

We argue that this minimal formalization in fact has sufficient expressive capacity to encode, in a reasonable intuitive way, all the kinds of semantic QoS attributes in the process model; we also argue that this minimal formalization has good properties with respect to creation, maintenance/updating and searching for annotation. We believe that these advantages are especially strong in the case of discourse annotations, because of the prevalence of the cross-cutting structures and the need to compare multiple annotations representing different purposes and perspective.

For evaluation the context QoS attributes in query process over the applicants process models we consider the membership functions representing the predicates interpreting the quality attributes in query process models, Table 2 summarizes the fuzzy modeling of numerical QoS attributes of interest.

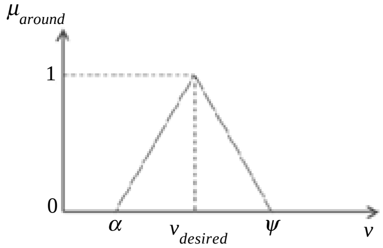
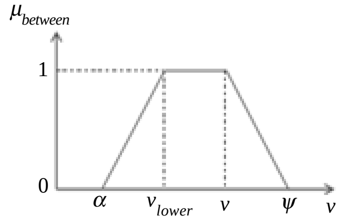
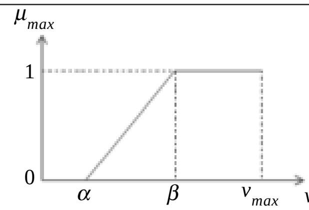
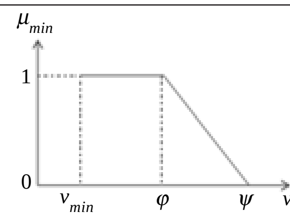
Numerical QoS attribute	Fuzzy Interpretation	
around $(t, v_{desired}, \mu_{around})$	$\mu_{around} = \begin{cases} 0 & r \le \alpha, v \ge \psi \\ \frac{v-\alpha}{v_{desired}-\alpha} & \alpha \prec v \prec v_{desired} \\ 1 & v = v_{desired} \\ \frac{\psi - v}{\psi - v_{desired}} & v_{desired} \prec v \prec \psi \end{cases}$	
between $(t, v_{low}, v_{up}, \mu_{between})$	$\mu_{between} = \begin{cases} 0 & r \le \alpha, r \ge \psi \\ \frac{r-\alpha}{r_{low}-\alpha} & \alpha \prec r \prec r_{low} \\ 1 & r_{low} \prec r \prec r_{up} \\ \frac{\psi - r}{\psi - r_{up}} & v_{desired} \prec v \prec \psi \end{cases}$	
max (t, μ_{max})	$\mu_{max} = \begin{cases} 0 & r \le \alpha \\ \frac{r-\alpha}{\beta-\alpha} & \alpha \prec r \prec \beta \\ 1 & \beta \prec r \prec r_{max} \end{cases}$	
min (t, μ_{min})	$\mu_{min} = \begin{cases} 1 & r_{min} \le r \le \varphi \\ \frac{\psi - r}{\psi - \varphi} & \alpha \prec r \prec \psi \\ 1 & r \ge \psi \end{cases}$	

Table 2. Fuzzy modeling of numerical QoS expressions

In our semantic QoS similarity measure we aim to calculate to which degree an annotation a: (t, v) in service applicant process models satisfies fuzzy underlying quality attribute expression q

For example, consider the modifier between. A fuzzy expression q: *between* $(m, v_{lower}, v_{upper})$ is characterized by the membership function $(\alpha, \beta, \varphi, \psi)$, $\beta = r_{low}$ where and $\varphi = r_{up}$; α and ψ are two values from the universe X. Let a: (t, v) be an annotation of an applicant graph, the similarity degree of quality attribute expression q according to a is given by:

q is completely satisfied if $v \in [v_{lower}, v_{upper}]$: $\delta(q, a) = 1$

more v is lower than v_{lower} or higher than v_{upper}, less q is satisfied: $0 \prec \delta(q, a) \prec 1$;

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

$\delta(q, a)$ is the semantic QoS similarity measure between expression q and annotation a

4. Linguistic quantifier-based approach for aggregating similarities measurement

In the section precedent we show how to measure the similarities between process models nodes, except structural similarity which are calculated over the overall workflow process model, the other similarities have been applied just for pairs from both process models query and applicant. So it remains to synthesize for each similarity criteria the overall similarity degree which

We have specified our similarities measures which are Syntactic similarity, Linguistic Similarity, Structural Similarity and Semantic Similarity; each similarity type covers a set of functionalities in the relation between the process model query and the applicant target. For example the syntactic similarity measures the degree of fulfillment between process models labels, the linguistic similarity come one step forward, it allow to approach synonym words and the phrasal labels supporting near meaning. However, syntactic and linguistic similarity measures by themselves do not exploit the structure of graphs. For example the user want to make a flight reservation and hotel booking but the process model is a package constituted from composition of flight reservation , hotel booking and car rent; thus will be seen in the corresponding process model structure, and the structural similarity degree will reveal the difference between the request and target process models.

Other important criteria in process models ranking is semantic QoS aspect of process models, Indeed semantic QoS makes the process models explicitly context-aware mainly on its quality attributes, so more this similarity degree is higher more semantic QoS attributes are fitted and vice versa.

In the next we present the quantifier based technique (using the quantifier predicate almost all) in the evaluation of our similarities degrees.

4.1 Theoretical Linguistic quantification

According to oxford dictionary a quantifier is an expression (e.g. *all, some*) that indicates the scope of a term to which it is. Linguistic quantifiers present linguistic expressions and make reference to a quantities, usually they are imprecise and their goal is to tolerate exceptions in aggregation of conditions, example of linguistic quantifiers are at least 2, almost 3, almost all…(Almost all conditions are satisfied)

For explaining how this method works, let X be an ordinary set of elements, R is a fuzzy predicate and Q is a linguistic quantifier, we want to calculate two index for the expression "*Q X are R* " which are the necessity and the possibility indexes. "*Q X are R*" signify "*within the elements of X there are Q satisfying R* " or again

"*Q X are R* " $\Leftrightarrow$ the fuzzy set of X elements satisfying "*Q are R* "

This means that X represent the possible set and it provide the possibility and the necessity of the event "*Q are R* "

Let µbe the membership function of the expression "*QX are R*", the truth degree $\mu(x)$ describe to each degree the fuzzy element x of X verify "*x is R* "

The possibility distribution π represents the truth degree given by the membership functions of X, and the function μ represent the fuzzy event "*Q are R* " calculated by OWA operator (see below).The overall possibility Π and necessity N of the fuzzy expression "*Q are P* " are:

$$\Pi = \max_{x \in X} (\min (\pi (x), \mu (X)) \text{ and}$$

$$\mathrm{N} = \mathop{\mathrm{mix}}_{x \in X} (\max (1 - \pi(x)), \mu (X))$$

More the necessity value N is high more there is lots of elements from X that satisfy R, we note that if no element of X satisfying R so the necessity is null.

It remains to introduce the calculation of the values of μ (representing the quantification "*Q are R*") using an OWA operator.

The interpretation of a quantified proposition of type "*Q X are R*" by an OWA operator [29] [30] is limited to monotones quantifiers (increasing or decreasing)

Q is increasing	Q is decreasing
$Q(0) = 0$	$Q(0) = 1$
$\exists k$ such as $Q(k) = 1$	$\exists k$ such as $a > b$ then $Q(a) \geq Q(b)$
$\forall a, b$ if $a > b$ then $Q(a) \geq Q(b)$	$\forall a, b$ if $a > b$ then $Q(a) \leq Q(b)$

A linguistic quantifier can be increasing (resp decreasing) which means that an increase in the satisfaction of R cannot decrease (resp increase) the truth value of the statement "*Q X are R* ", at least 3, almost all (resp at most 2, at most the half) are examples of increasing (resp decreasing examples).

Let be $X = \{x_1, x_2, ..., x_n\}$ such as $\mu_A(x_1) \geq \mu_A(x_2) \geq ... \geq \mu_A(x_n)$. The interpretation of "*Q X are R*" with Q is increasing is given by:

$$OWA = \sum_{i=1}^{n} (w_i \times \mu_A(x_i))$$

And since Q is relative, we have $w_i = \mu_Q\left(i/n \right)$. Considering the membership function, μ_Q, defining the linguistic quantifier "*almost all* ", s.t. $\mu_Q\left(i/n \right) = i/n$

Every weight w_i represent the growing of the degree of truth of the quantification if we compare a situation where there is exactly $(i-1)$ elements completely R with a situation involving (i) elements entirely R (the other elements absolutely does not satisfy R).

4.2 Linguistic quantification of similarities measurements

In the section above we have introduces the theoretical calculation of statements of type "*Q X are R* ". Now, returning to our similarities measurements, we aggregate our atomic similarities values by using the linguistic quantifier almost all. The table 4 gives the natural interpretation of the similarities between request *r* and applicant *a* process model

Type of similarity	Natural Interpretation
Syntactic similarity SYN-Sym	"Almost all nodes of *r* have the same syntax in *a*"
Linguistic similarity LIN-Sym	"Almost all nodes of are linguistically similar to nodes in *a*"
Semantic similarity SEM-Sym	"Almost all semantic preferences of *r* are satisfied by *a*"

Table 3. Similarities criteria quantification by "*almost all* "

In the scenario example, when he arrives to the airport, Omar wants to make hotel and car booking, figure 15 shows his request indicating he demands a service package containing Hotelfinding, Hotelbooking and carRenting services, the workflow graph associated to context query is therefore annotated with Omar quality attributes in hotel place and cost and also car renting cost and car reliability, the request process model is matched with a service applicant process model annotated with some valuated attributes which describe quality of service and other relative functional and non-functional semantic attributes.

In this illustration we exemplify the calculating of the established similarities measures. The mapping between the two process models depends first on the structural similarity which takes into account node's position (i.e. control ûow, predecessor and successor nodes) and which includes into its calculation process the syntactic similarity measurement. Secondly we interest to the semantic QoS similarity which depends on Linguistic similarity.

• Syntactic Similarity measure:

At first, for calculating syntactic similarity we use the established method $SYN - Sym$ to calculate syntactic similarity between labels nodes as shown in table 4

n_i	n_j	SYN-Sym (n_i, n_j)
bookFlight	bookingFlight	0, 76
SearchHotel	SearchingHotel	0, 78
BookHotel	Booking	0, 44

Table 4. Syntactic similarity measures

The statement "*almost all nodes of r have the same syntax in a*" have the form "*Q X are R* " with Q is the quantifier "*almost all*" and *X* is the set of labels pairs from request and applicant process models, SYN-Sym is the membership function of the syntactic similarity metric.

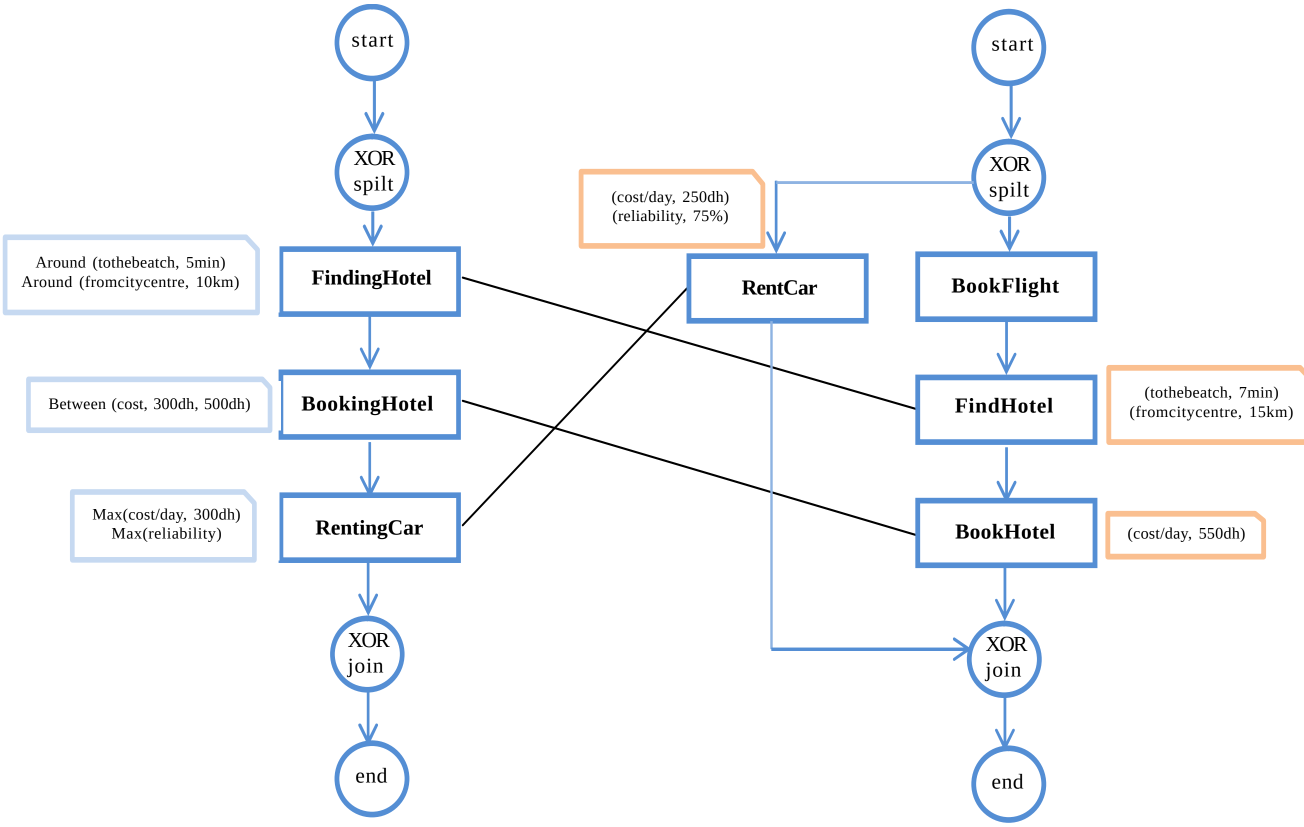


Figure 15. Request and applicant process models matching

Basing on OWA operator, for calculating overall syntactic similarity we rank order in decreasing order the atomic syntactic similarities, we get:

SYN-Sym (SearchHotel, SearchingHotel) = 0.78 >= SYN-Sym (BookFlight, BookingFlight) = 0.76 >= SYN-Sym (BookHotel, Booking) = 0.44

After, we calculate the overall syntactic similarity degree: SYN-Sym (r, a)

$$= \max (\min (0.78, \mu_Q\left(1/3\right), \min (0.76, \mu_Q\left(2/3\right), \min (0.44, \mu_Q\left(3/3\right) = 0.66.$$

SYN-Sym (r, a) = max (min (0.78, 0.33), min (0.76, 0.66), min (0.44, 0.99) = 0.66

This means that at least 66%of nodes are syntactically similar to at least a degree 0.66. Effectively, we observe that just 2node among 3 have a similarity degree greater than 0.75 which is greater than or equal to 0.66.

• QoS Semantic Similarity measure:

In order to evaluate semantic QoS attributes, we consider from Figures 15 the pairs, (r_i, a_j), such that r_i is a request attribute node i and a_j is its corresponding annotation in applicant a as shown in Table 5.

After defining the membership function of each semantic QoS expression presented in Table 2, the truth degree of each expression r_i is computed as shown in Table 6.

Basing on OWA operator, for calculating overall semantic QoS similarity we rank order in decreasing order the atomic semantic

p_i	a_j	Pref-Eval (p_i , a_i)
p_1: Around (tothebeatch,5min)	a_1: (tothebeatch,7min)	$QSemantic - Eval\ (r_1, a_1)$
p_2: Around (fromcitycenter,10km)	a$_2$: (fromcitycenter,15km)	$QSemantic - Eval\ (r_2, a_2)$
p_3: Between(cost,300dh,500dhs)	a_3: (cost/day,550dh)	$QSemantic - Eval\ (r_3, a_3)$
p_4: Max(cost/day,300dh)	a_4: (cost/day,250dh)	$QSemantic - Eval\ (r_4, a_4)$
p_5: Max(reliability)	a_5: (reliability,75%)	$QSemantic - Eval\ (r_5, a_5)$

Table 5. QoS semantic similarity elements

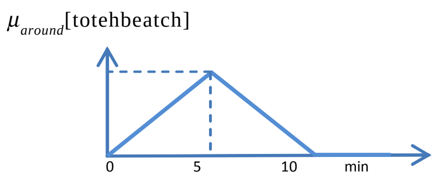
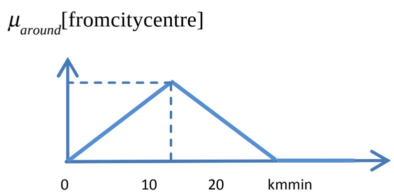
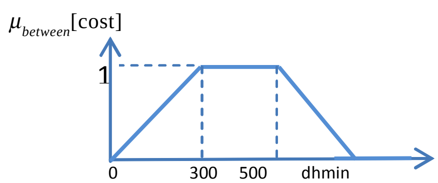
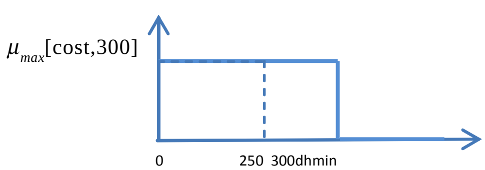
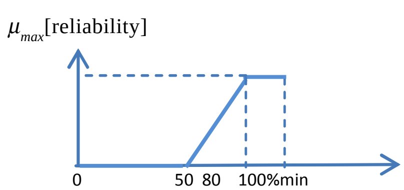
Preference p_i	Membership function of p_i	Pref-Eval (p_i , a_i)
p_1		$QSemantic - Eval\ (r_i, a_i) = 0{,}65$
p_2		$QSemantic - Eval\ (r_i, a_i) = 0{,}5$
p_3		$QSemantic - Eval\ (r_i, a_i) = 0{,}9$
p_4		$QSemantic - Eval\ (r_i, a_i) = 1$
p_5		$QSemantic - Eval\ (r_i, a_i) = 0{,}85$

Table 6. QoS semantic similarity computing between matched pairs

QoS similarities degrees, we get: Pref-Eval (p_4, a_4) = 1 ≥ Pref-Eval (p_3, a_3) = 0,9 ≥ Pref-Eval (p_5, a_5) =0,85 ≥ Pref-Eval (p_1, a_1) = 0,65 ≥ Pref-Eval (p_2, a_2) = 0,5.

After, we calculate the overall similarity degree: Pref-Eval (r, a)

= max (min (1, $\mu_Q\left(1/5\right)$, min (0,9, $\mu_Q\left(2/5\right)$, min (0,85, $\mu_Q\left(3/5\right)$, min (0,65, $\mu_Q\left(4/5\right)$, min (0,5 $\mu_Q\left(5/5\right)$ = 0.65.

Pref-Eval (r, a) = max (min (1, 0.2), min (0.9, 0.4), min (0.85, 0.6), min (0.65, 0.8), min (0, 5, 1).

This means that at least 65%of nodes are semantically similar to at least a degree 0.65. Effectively, we observe that 3 nodes among 5 (i.e.3/5= 0.6) have a similarity degree greater than 0.75 which is greater than or equal to 0.65.

For the other two similarities, Structural similarity is not aggregated by the linguistic quantifier because the value of overall structural similarity is directly calculated its formula (see the structural similarity computing section), and as regard to linguistic similarity, its calculating by the quantifier "*almost all*" using WordNetfor atomic linguistic comparison.

5. Discuss about process models ranking

Previous section presented a quantified based method to calculate the overall similarity degree between process models. We get the set of process models applicants resulting from discovery process and we construct the process model request based on semantic QoS attributes of the context. In this section we discuss some methods to rank-order process models based on their similarities to the request process models. However, in our similarities criteria set, there is two types of criteria: criteria that are more oriented to mapping process models which are structural similarity and syntactic similarity and other serves more to semantically differentiate process models which represent the linguistic and the semantic QoS similarities.

In general, two families of methods to rank-order process models exist, methods basing on aggregation and methods without aggregation.

5.1 Methods basing on aggregation

To rank-order the process models we first compute the mapping degree aggregating the structural and syntactic similarities measures: $Mapp - \deg = w_1 \times STR - Sym(r, a) + (1 - w_1) \times SYN - Sym(r, a)$ where $0 < w_1 < 1$ is a weight assigned to the mapping degree aggregating both structural and syntactic similarities criterions.

On the other hand, similar to mapping degree, we calculate the semantic satisfaction degree $SEM - \ Satisfaction = w_2 \times SEM - Sym(r, a) + (1 - w_1) \times LIN - Sym(r, a)$ with $0 < w_1 < 1$ is a weight accorded to the semantic degree aggregating both semantic QoS and linguistic similarities criteria.

The decision making based on the two indices: mapping degree and semantic satisfaction degree depend intrinsically to the user/agent activity and the effectiveness of the four similarities measurements, here we presents two methods to do so:

• The first is to average both mapping and semantic degrees either with weighting them or without, thus the resulting degree handles both criteria, and the ranking is based on this overall degree.

• The second is to consider a rate success for the mapping degree, so if it reaches some degree it is acceptable mapping and then we rank-order by only the semantic degree satisfaction, if not we consider that the mapping is not reachable and thus we neglect the process model.

5.2 Methods without aggregation

The two distinct similarity degrees are used to rank-order applicant process models. The answers are ranked by using the lexicographical order. A priority is given to the mapping degree while the semantic satisfaction is only used to break ties.

6. Conclusion

In this paper we use the services and request process models for representing both their quality of service and behavior aspects for selecting the best-fitting service, thus we take advantage of the makeover of process models matchmaking problem to graphs comparing problem, it follows from this, firstly we leverage from the syntactic as semantic wealth that allow workflow-based representation in query and service description, secondly we perform our four similarities measures based on the graphs mapping and thirdly we have used a quantification approach for aggregating the calculated four atomics criteria of similarity. In the future works of our researches we aim to continue developing our approach by implementing our selection approach in large spectrum of mobile and context aware situations, also we seek to develop our mechanism by adding other similarities measures basing on workflow comparison.

References

[1] Mark Weiser. (1991). The computer for the 21st century, *Scientific American*, 66.

[2] Sodki Chaari, Enhancing Web Service Selection by QoS-Based Ontology and WS-Policy.

[3] Kim, Y., Doh, D. (2007). A trust type based model for managing QoS in web services com-position. *International Conference on Convergence Information Technology*, 0:438-443.

[4] Lee, K., Jeon, J., Lee, W., Jeong, S., Park, S. (2003). QoS for web services: Requirementsand possible approaches. Technical report, W3C, *Web Services Architecture WorkingGroup*, November.

[5] Ran, S. (2003). A model for web services discovery with QoS. *ACM SIGecom Exchanges*,4 (1) 1-10.

[6] Yu, W. D., Radhakrishna, R. B., Pingali, S., Kolluri, V. (2007). Modeling the measurements of QoS requirements in web service systems. *Simulation journal*, 83 (1) 75-91.

[7] Rosenberg, F., Platzer, C., Dustdar, S. (2006). Bootstrapping performance and dependabilityattributes of web services. pages 205–212. *IEEE Computer Society*.

[8] Garcia, D. Z. G., de Toledo, M. B. F. (2006). Semantics-enriched qos policies for web serviceinteractions. *In*: Proceedings of the 12th Brazilian Symposium on Multimedia and the web, p. 35-44.

[9] Menasce, D. A. (2002). QoS issues in web services. *IEEE Internet Computing*, 6 (6) 72-75, Nov/Dec.

[10] Yu, W. D., Radhakrishna, R. B., Pingali, S., Kolluri, V. (2007). Modeling the measurements ofQoS requirements in web service systems. *Simulation journal*, 83 (1) 75-91.

[11] Rahman, W., Meziane, F. (2008). Challenges to describe QoS requirements for web servicesquality prediction to support web services interoperability in electronic commerce. *In*: Proceedings of the 10th IBIMA Conference on Innovation and Knowledge Managementin Business Globalization, Kuala Lumpur, Malaysia, 30 June - 2 July, 4 (6) 50-58.

[12] Self, J. (1999). The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. *International Journal of Artificial Intelligence in Education*, 10, 350-364.

[13] Schleno, C. etc. (2000). The Process Specification Language (PSL): Overview and Version 1.0 Specification. Technical Report NISTIR 6459, *National Institute of Standards and Technology*, Gaithersburg, MD.

[14] Narayanan, S. (1999). Reasoning about actions in narrative understanding. *In*: International Joint Conference on Artificial Intelligence (IJCAI'1999), p. 350-357. Morgan Kaufmann Press.

[15] Levesque, H. J., Reiter, R., Lesperance, Y., Lin, F. Z., Scherl, R. B. (1997). GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31 (1-3) 59-83.

[16] Martin, D., Cheyer, A., Moran, D. (1999). The Open Agent Architecture: a framework for building distributed software systems. *Applied Artificial Intelligence*, 13 (1/2) 91-128.

[17] van der Aalst, W. M. P., Kees van Hee. (2002). Workflow Management - Models, Methods, and Systems. MIT Press.

[18] Andreas Wombacher, Peter Fankhauser, Bendick Mahleko, Erich Neuhold, (2004).Matchmaking for Business Processes Based on Choreographies. *Intl. Journal of Web Services*, 1 (4) 14-32.

[19] James L. Peterson. (1981). Petri Net Theory and the Modeling of Systems. Prentice-Hall.

[20] Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8 (3) 231-274, June.

[21] Dumas, M., García-Bañuelos, L., Polyvyanyy, A., Yang, Y., Zhang, L. (2010). Aggregate quality of service computation for composite services. *In*: Maglio, P. P., Weske, M.,Yang, J., Fantinato, M. (eds.) ICSOC. *Lecture Notes in Computer Scienc*e, 6470, 213-227

[22] Levenshtein, I. (1966). Binary code capable of correcting deletions, insertions and reversals. *Cybernetics and Control Theory*, 10 (8) 707-710.

[23] Remco M. Dijkman, Marlon Dumas, Luciano García-bañuelos. (2009). Graph Matching Algorithms for Business Process Model Similarity Search, *In*: Proceeding of the Conference: Business Process Management - BPM , p. 48-63.

[24] Miller, G. (1995). Wordnet: A lexical database for english, *Communications of the ACM*, 38 (11) 39-41.

[25] Shapiro, L. G., Haralick, R. M. (1985). A Metric for Comparing Relational Descriptions, *IEEE Trans. Pattern Analysis and MachineIntelligence*, 7 (1) 90-94, Jan.

[26] Eshera, M.A., Fu, K.S. (1986). An Image Understanding SystemUsing Attributed Symbolic Representation and Inexact Graph-Matching, *J. Assoc. Computing Machinery*, 8 (5) 604-618.

[27] Sanfeliu, A., Fu, K. S. (1983). A Distance Measure between AttributedRelational Graphs for Pattern Recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, 13 (3) 353-362, May.

[28] Bunke, H., Shearer, K. (1998). A graph distance metric based on themaximal common subgraph, *Pattern Recognition Letters*, 19, 255–259.

[29] Yage, R. R. (1993). Families of OWA operators, *Fuzzy Sets and Systems*, 59, 125-148.

[30] Bosc, P., Liétard, L. on the extension of OWA operator to evaluate some quantifications, *In*: Proceeding of the european Congress on fFuzzy and Intelligent Technologies (EUFIT'93), Aachen (Germany), 332-338.