Journal of Universal Computer Science, vol. 16, no. 1 (2010), 198-217
submitted: 1/2/09, accepted: 15/10/09, appeared: 1/1/10 © J.UCS

LCP-Nets: A Linguistic Approach for Non-functional Preferences in a Semantic SOA Environment

Pierre Châtel
(Thales Communications France
1-5 avenue Carnot, Massy, 91883, France
pierre.chatel@thalesgroup.com)

Isis Truck
(LIASD – EA 4383, Université Paris 8
2 rue de la Liberté, Saint-Denis Cedex, 93526, France
truck@ai.univ-paris8.fr)

Jacques Malenfant
(Université Pierre et Marie Curie-Paris 6, CNRS, UMR 7606 LIP6
104 av. du Président Kennedy, Paris, 75016, France
Jacques.Malenfant@lip6.fr)

Abstract: This paper addresses the problem of expressing preferences among non-functional properties of services in a Web service architecture. In such a context, semantic and non-functional annotations are required on service declarations and business process calls to services in order to select the best available service for each invocation. To cope with these multi-criteria decision problems, conditional and unconditional preferences are managed using a new variant of conditional preference networks (CP-nets), taking into account uncertainty related to the preferences to achieve a better satisfaction rate. This variant, called LCP-nets, uses fuzzy linguistic information inside the whole process, from preference elicitation to outcome query computation, a qualitative approach that is more suitable to business process programmers. Indeed, in LCP-nets, preference variables and utilities take linguistic values while conditional preference tables are considered as fuzzy rules which interdependencies may be complex. The expressiveness of the graphical model underlying CP-nets provides for solutions to gather all the preferences under uncertainty and to tackle interdependency problems. LCP-nets are applied to the problem of selecting the best service among a set of offers, given their dynamic non-functional properties. The implementation of LCP-nets is presented step-by-step through a real world example.

Key Words: preference modelling, fuzzy linguistic approach, CP-nets, Web service filtering
Category: H.3, J.0

1 Introduction

Service-oriented architectures (SOA) deal with the growing need for distributed applications capable of evolving continuously over their execution. In the context of the Web, atomic feature producers are called Web services, and consumers

Web processes. These Web services can appear and disappear at runtime, thus requiring a loose coupling between service providers and consumers. We adopt an SSOA (Semantic Service-Oriented Architecture) approach where usual service offers are enhanced with high level business concepts extracted from ontologies as well as non-functional commitments, and are published in dedicated service registries. Based on this information, a loose coupling is implemented at runtime by the late-binding of an abstract service request (originating from the orchestration of a Web process) to a concrete service offer.

This paper focuses on the various steps involved in the implementation of the ultimate part of this matchmaking process: the dynamic selection of Web services based on non-functional consumer preferences and up-to-date QoS (Quality of Service) values of monitored services. In this multi-criteria decision making context, to obtain a total order over services and make specific binding decisions between consumers and producers, we propose a solution based on preferences established among the various non-functional properties of required services. Given the subjective nature of these preferences, they are elicited by business process programmers before running them in the SSOA framework.

The idea is to imagine an approach to elicit and exploit consumers preferences expressed qualitatively, i.e. with linguistic concepts.

In the rest of the paper, first the conceptual and technical framework is introduced, then we briefly describe a set of tools for representing and reasoning with conditional preference statements: the "*CP-nets"[1]. Section 3 introduces our proposal of using a fuzzy linguistic variant of CP-nets in modeling preferences, Section 4 gives a proof-of-concept example, while Section 5 points out conclusions and future work.

2 Background and Related Work

This work takes place in the SSOA and Web services environments that are introduced below. We then briefly detail the related work concerning the fuzzy linguistic approach and the formalism of *CP-nets.

2.1 SSOA and non-functional properties

SOA and Web services have recently gained broad industry acceptance as established standards. They provide for greater interoperability and some protection from lock-in to proprietary vendor software. However an SOA can be implemented using any kind of service-based technology.

In our framework, two distinct roles are identified. *Service providers* implement (generic) functionalities made available to applications as *Web services*,

[1] By *CP-nets we mean any kind of Conditional Preference Networks (the asterisk substitutes as the wildcard), i.e. CP-nets, TCP-nets, UCP-nets, etc.

thanks to SOA standards like, e.g., service registries. *Service consumers* request and use services available on the network according to their specific requirements through service invocations made by *business processes*.

To cope with the dynamism of the Web, the binding of Web services (from providers) to business processes (of consumers) is established on the fly, at runtime. To achieve this, and to provide for high interoperability among heterogeneous service offers and requests, we make this binding go much further than the traditional syntactic approach by first using semantic annotations on service offers and requests to identify offers that match each request. This higher level of abstraction is complementary to the usual syntactic definition.

In our approach, semantics encompasses not only functional (what services do and processes need) but also non-functional properties (QoS and related properties). Functionality of service offers concerns the core business work provided by each Web service. We strive to semantically describe the main features of the service: its end goal and the ontological concepts associated to its input and output parameters. We use SAWSDL [Lausen and Innsbruck, 2007], an extension to the WSDL service interface definition language that introduces semantic annotations, to annotate otherwise syntax-based service offers with classes (i.e. concepts) from domain ontologies. Service requests express similarly their semantic requirements for the completion of their task. Non-functionality, on the other hand, concerns the level of QoS guaranteed by Web services and required by business processes. For each service call, functionally matching offers from registries are further filtered to keep only the ones which non-functional commitments fulfill the requirements of the calling business process.

But we also strive to build Quality of Service (QoS) awareness into the *runtime* SSOA platform to dynamically select the best service(s) available to fulfill each request. Indeed, after filtering with functional and non-functional constraints, we use non-functional information again to further seek the best offer(s) just prior invoking the service. Hence, at runtime, non-functional requirements become preferences applied to currently measured QoS values associated to the statically filtered services. These concepts are being implemented in the SemEUsE[2] ANR project, as seen in Figure 1, where the various components involved in service filtering and selection are shown alongside a simple example: five available services from the registry are filtered then the remaining ones (*S1* and *S3*) are dynamically selected upon given their current QoS values and statically defined preferences.

A major issue when dealing with QoS is the large number of different dimensions (e.g. latency, precision, etc.) of importance. Because one rarely gets an offer that is the best for every different QoS dimension, we need consumer preferences to rank offers given their relative strength on the different dimen-

[2] `http://www.semeuse.org`

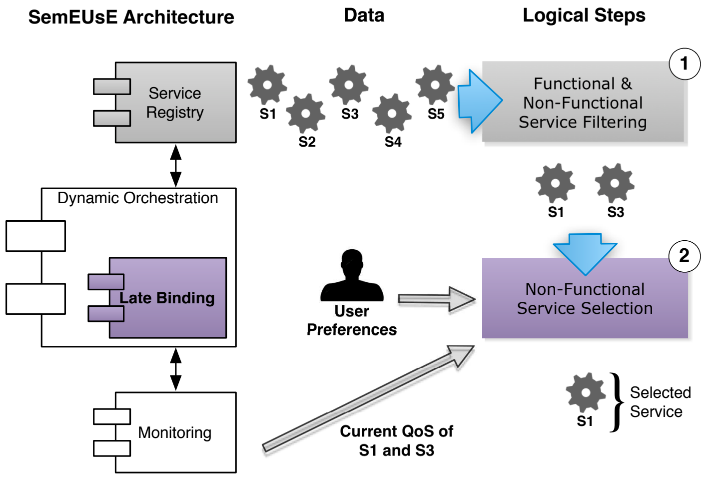

Figure 1: SemEUsE service selection architecture.

sions. Preference elicitation and expression have received attention in past years and several formalisms to do so have been proposed. In the context of SOA, a good formalism must obey several requirements, among which usability by non-specialists, like business process programmers, is of primary importance. To this end, we propose a new formalism based on the combination of CP-nets and the fuzzy linguistic approach [Zadeh, 1975] to qualitatively specify the preferences of Web consumers over the different non-functional properties of offers.

2.2 Fuzzy Linguistic Approach

The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables [Zadeh, 1975]. Appropriate linguistic descriptors must be chosen to form the term set as well as their semantics. The universe of the discourse over which the term set is defined can be arbitrary. In this paper, we shall use the interval $[0, 1]$. Odd cardinality term sets, typically 5, 7 or 9, are preferred [Delgado et al., 1993, Herrera and Martínez, 2000], representing the mid term by an assessment of "approximately 0.5", other terms being placed symmetrically around it. For example, a set of five terms T, could be given as: $T = \{s_0 : \textit{very low}, s_1 : \textit{low}, s_2 : \textit{medium}, s_3 : \textit{high}, s_4 : \textit{very high}\}$. It is also required that there exist negation *Neg*, *max* and *min* operators defined over this set [Herrera and Martínez, 2000]: (i) a negation operator $\text{Neg}(s_i) = s_j$ such that $j = g - i$ ($g+1$ is the cardinality), (ii) a max operator: $\max(s_i, s_j) = s_i$ if $s_i \geq s_j$, (iii) a min operator: $\min(s_i, s_j) = s_i$ if $s_i \leq s_j$.

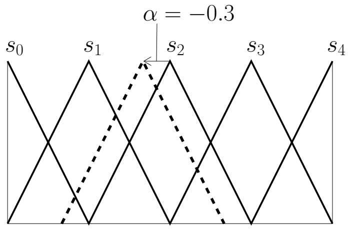

Figure 2: Lateral displacement of a linguistic label $\Rightarrow$ $(s_2, -0.3)$ 2-tuple.

The semantics of the terms is given by membership functions. Linear trapezoidal (even triangular) functions are often considered good enough to capture the vagueness of those linguistic assessments. The use of linguistic variables implies the processes of computing with words for their fusion, aggregation, comparison, etc. To perform these computations, different models have been used such as the semantic [Degani and Bortolan, 1988], the symbolic [Delgado et al., 1993, Truck and Akdag, 2006] or the 2-tuple [Herrera and Martínez, 2000] representation models. Given a linguistic term, the 2-tuple formalism provides for a pair (fuzzy set, symbolic translation) = (s_i, α_i) with $\alpha_i \in [-0.5, 0.5)$ as can be seen in Figure 2 where the obtained 2-tuple is $(s_2, -0.3)$. The computational model based on linguistic 2-tuples carries out processes of computing with words easily and without loss of information.

2.3 *CP-nets

*CP-nets designate a family of well-known graphical formalisms used for the expression of user preferences and naturally suited when preferences are easily approximable by lexicographic rules. We have chosen to ground our non-functional preference modeling on these formalisms since they exhibit specific benefits like ease of use for the preference modeler, relatively low computation cost, are strongly structured and could be easily extended to support additional properties needed in our context. Other formalisms exist in the literature such as GAI (Generalized Additive Independence) networks [Gonzales et al., 2008] but these approaches are not really suitable because of the elicitation effort they imply. Indeed they are useful when discriminating among a huge quantity of possibilities, which is not the case here.

A **CP-net** (conditional preference network) is a compact graphical representation of qualitative user preferences [Boutilier et al., 2004]. It is relatively intuitive. Its main elements are: *nodes* representing the problem variables, *arcs* denoting preferences among these variables for given values, and conditional preference tables or *CPTs*. CPTs express the preferences over values taken by variables, defining in extension the binary relationship between them. CP-nets allow for the preference modeling of statements such as "I prefer the V_1 value for property X over V_2 if properties Y equals V_Y and Z equals V_Z". In fact, this graphical representation allows one to express the dependency between connected CPTs. Hence preferences can be expressed conditionally to the values taken by their parent nodes in the graph, but regardless of the values taken by the other nodes (this is the *ceteris paribus* property, central to CP-nets, meaning "*all other things being equals*"). There is also a notion of relative preference between the preferences themselves: a CPT associated with a specific node has a higher priority than the CPTs of its offspring. This notion of relative preference is taken into account when globally comparing complete assignments (tuples of values binding all of the preference variables). Most of the inference computations and logic reasoning that can be made on a CP-net are practicable from an algorithmic complexity point of view when this CP-net obeys some restrictions. The major restrictions pertaining to this framework are the generalized use of acyclic graphs, the limited use of the indifference relationship modeling preferences (neither explicitly better nor worse) among variables, and therefore the systematic definition of total pre-orders in the CPTs for each distinct parent node [Boutilier et al., 2004].

Utility CP-nets, or **UCP-nets** [Boutilier et al., 2001], differ from CP-nets by replacing the definition in extension of the binary relationship between node values in CPTs by numerical utility factors. Doing so, node values may retain their qualitative form: only preferences are quantified with utility values. This shift is motivated by the fact that the precision of a utility function (as opposed to a preference ordering) is often needed in decision making contexts where uncertainty is a factor. It is also motivated by the fact that a CP-net does not allow for the comparison or the ordering of all its alternatives. This limitation is also solved by the quantification of the preferences [Boubekeur and Tamine-Lechani, 2006]. A utility factor is a real number associated to an assignment of a node X from the network, given a specific assignment of its parent nodes. Utility factors express preference degrees for the different assignments. In UCP-nets, preference modelers use CPT utility factors to deliver their preferences local to the variables of this CPT, but rely on the UCP-net semantics to compute the global utility of each complete assignment to enable their comparison.

Another extension of CP-nets, named Tradeoffs-enhanced CP-nets, or **TCP-nets** [Brafman and Domshlak, 2002], allows one to express preferences of the form: "A better assignment for X is more important than a better assignment for Y". These are called *relative importance* statements. TCP-nets also generalize this class of preferences in order to accept *conditional relative importance* statements. With these, it becomes possible to express preferences of the form: "A better assignment for X is more important than a better assignment for Y given that $Z = z$". This formalism introduces a new kind of preference tables, the "Conditional Importance Tables" (or CIT), as well as two new types of arcs between nodes: *i-arcs* and *ci-arcs*. These arcs allow, respectively, for the modeling of *basic* and *conditional* relative importance statements. Basically, TCP-nets empower users to express the tradeoffs they are willing to concede between various preference criterions, given the current assignment to preference variables. The notion of conditional relative importance complements the one of conditional *ceteris paribus* independence in order to provide for a richer conceptual framework to model and reason about user preferences.

The idea of mixing CP-nets and non-functional properties has already been addressed by Schröpfer *et al.* [Schröpfer et al., 2007]: the authors define preferences through CP-nets to select the best service in an SOA. But they don't consider qualitative preferences nor continuous domains of variables. They only tack on the CP-nets formalism to their preference modeling.

3 Fuzzy Linguistic Approach and CP-nets

*CP-nets exhibit two important limitations to express preferences in a QoS setting. Many QoS dimensions are defined on continuous domains, but *CP-nets only deal with finite domain variables. We propose to discretize continuous domains using fuzzy linguistic terms [Zadeh, 1975] instead of crisp sets. In a context where users have to express preferences among values of continous domains, e.g. latencies, the qualitative nature of fuzzy sets with smooth transitions proved to better capture users intention. Hence, this shall allow for a better service selection when two services have properties like "latency" that have more or less similar values, by avoiding to artificially put large differences in preferences between otherwise nearby domain values. Another problem is that precise utility values are hard to get from non-specialist users. Indeed, giving numbers to express a perception (or a preference in our case) is not always feasible. Facing this, current *CP-net models provide for only two alternatives. On the one hand, the original CP-net model expresses preferences through a simpler and more intuitive order relation (without precise utility values) but suffers from lower performance when comparing two assignments. On the other hand, UCP-nets allow for such a comparison to be performed quite efficiently but with much harder-to-get precise numerical utility values.

Another proposition in this paper is to express utility values qualitatively, i.e. using words translated into fuzzy sets. As in UCP-nets with numerical utility values, comparisons between assignments use the global utility of these assignments, but it will be computed using fuzzy logic and aggregation tools.

3.1 Linguistic CP-nets (LCP-nets)

In [Châtel et al., 2008] we have proposed a new variant of CP-nets, called LCP-nets (Linguistic Conditional Preference networks), to get the advantage of the fuzzy linguistic approach into a marriage of UCP-nets and TCP-nets. Compared to the previous *CP-nets, this new formalism allows for the preference modeling of more qualitative statements such as "I prefer the *more or less* V_1 value for property X over *exactly* V_2 if properties Y equals *approximately* V_Y and Z equals *a bit more than* V_Z". Moreover, these statements that resemble elaborated fuzzy rules are interpreted in a context where the overall preference on X shall take into account every such preference statement that applies, to some degree, to the value of Y.

The following constraints and properties from the SSOA context had to be taken into account:

– preferences must be easy to define since business process programmers can't rely on preference-modeling experts, nor do they have much resources to allocate for their elicitation. It implies that some imprecision must be tolerated in preference models,

– typical problems to be dealt with use few variables (commonly in the order of 10 variables),

– computation time for decision-making based on preferences can be seen as relatively small compared to the subsequent service invocation over the network.

It leads to the following sought-after properties of the preference formalism: it is **graphical** to ease their definition without overly compromising computation, since LCP-net models are much easier to establish than writing several sets of fuzzy rules that can be interdependent, and they are **qualitative** to deal with user or QoS sensor imprecision.

Due to the latter, linguistic variables [Zadeh, 1975] have been incorporated into LCP-nets, the semantic of each linguistic term being given by a fuzzy set. Thus, preference modelers can easily manipulate pre-existing linguistic terms during elicitation. Also, as in other graphical models, LCP-nets have nodes corresponding to problem variables which continuous domains are discretized as linguistic term sets. In the SSOA context, these variables refer to non-functional properties of services.

To sum up, LCP-nets allow users to express tradeoffs among variables using i-arcs or ci-arcs from TCP-nets and have CPTs similar to the ones of UCP-nets, but express utilities with linguistic terms rather than numerical values. With LCP-nets, it is possible to:

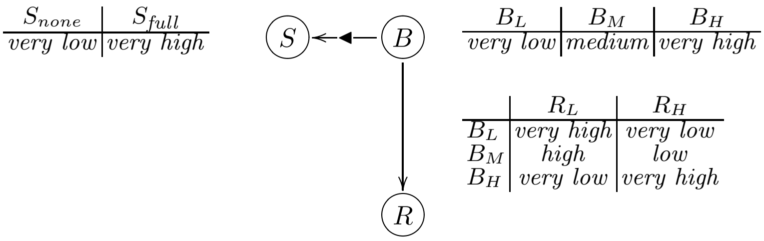

Figure 3: The imaging Web service QoS preferences example using LCP-nets.

– **elicit preferred assignments for a specific QoS domain (or interdependent ones)**, using CPTs similar to the ones of UCP-nets [Boutilier et al., 2001],

– **reveal relative importance of non-functional properties**, using arcs from CP-nets,

– **indicate tradeoffs between non-functional properties**, using i-arcs from TCP-nets [Brafman and Domshlak, 2002].

The aforementioned properties are illustrated in Figure 3, where user preference on the selection of an Imaging Web service (e.g. a security camera) is detailed. The overall goal of the user is to get images as fast as possible. This goal is translated into preferences according to three of its QoS properties: security (S), bandwidth (B) and image resolution (R). The user always prefers bandwidth over security, and if the bandwidth is low, she prefers low-resolution images to get them as fast as possible. More details on this preference model will be given in Section 4.

While our last paper [Châtel et al., 2008] stressed only the formalism (and the framework) of LCP-nets, we focus in this study on their underpinnings, i.e. the inference process behind the formalism.

3.2 LCP-nets framework: from elicited preferences to service selection

In the context of SSOA, the following conceptual steps are executed sequentially in order to implement the matchmaking process: a dynamic selection of Web services based on non-functional consumer preferences and up-to-date QoS values of monitored services. In particular, the preference representation and evaluation steps have been implemented in Java (using the *jfuzzylogic*[3] library for the fuzzy part) as an LCP-net support framework.

[3] http://jfuzzylogic.sourceforge.net

Indeed, one of the key aspects of LCP-nets lies in the representation and evaluation of its models. While the formalism itself has been inspired by *CP-nets, its tooling introduces another approach to the *CP-nets family, more specifically a mapping from a graphical model to a representation through multiple *Fuzzy Inference Systems* (or FIS) at compile-time. Evaluation of an LCP-net model is then based on its associated FIS, using the fuzzy-logic theoretical setting. The main idea behind this representation system translation is to gain the same level of flexibility at evaluation time than during preference elicitation: for instance, inputs of an LCP-net model can be crisp or fuzzy values over the defined domains, despite the strict use of linguistic terms in the Conditional Preference Tables.

3.2.1 Preference elicitation

As previously indicated, preference retrieval is conducted before runtime. But LCP-net preferences tell how to select services given their run-time QoS, allowing users to easily tailor Web process executions to each deployment scenario.

For example, in order to implement a pre-existing well-know process in the fire-fighting domain, dynamic service selection can be adjusted according to two of the possible deployment contexts: usual civil fire or crisis-management (where short intervention delays could be preferred over equipment capacities if the scene is distant).

3.2.2 Preference model translation

Backed by fuzzy logic, it is possible to translate preference models to an efficient decision-making representation used at runtime. In the process, each utility table in a preference model will be mapped to a single fuzzy rule set, as in fuzzy control [Driankov et al., 1993], to become a local FIS.

The inputs of these node-bound FIS can be crisp or fuzzily measured values obtained from monitored Web services.

3.2.3 Preference model evaluation

In the following, the preference model evaluation process is broken down into four key steps for selection of the best-suited service at runtime.

First, during **QoS value injection**, multiple QoS values are retrieved from a monitoring component of the SSOA framework or directly from the Web services themselves. These values can be of two kinds: crisp QoS values seen as "singleton" fuzzy sets of the considered QoS domain, or fuzzy QoS values that may need to be "adjusted" by a domain normalization on the [0,1] scale.

After QoS values have been retrieved, **local utility value inference** is launched. This inference of local node-bound utility values is made using Zadeh's Generalized Modus Ponens.

Currently, the fuzzy inference mechanisms are set once and for all in the implementation of LCP-nets, but some control could be given to the user over these, with the *caveat* that a good knowledge of fuzzy inference is needed to perform an enlightened choice. Another variant would call for an end-to-end linguistic treatment of this fuzzy inference; an approach that could better match user's intentions in expressing her preferences. Indeed 2-tuples that have been introduced in section 2.2 deal with linguistic statements without loss of information. Considering the preferences and the values for properties as 2-tuples, an *ad hoc* inference process [Alcalá et al., 2007] should then be used instead of Zadeh's Generalized Modus Ponens.

Finally, during **global utility value computation**, aggregation of previously inferred local utilities is made. But, during aggregation, we have to take into account the fact that arcs in preference models give the relative importance to their nodes and attached local utilities. For instance in the previously mentioned preference model (see Figure 3), bandwidth is more important than both security and resolution, these last two being of equal importance.

In UCP-nets, the numerical nature of utility values allows the user to express this relative importance by tuning the order of magnitude of utility values among the different tables. With linguistic terms, the user no longer has this possibility. In order to get this implicit relative importance back in LCP-nets when computing the global utility value, weights are associated to each node in the preference graph by a **weight computation process** correlated to their depth in the graph. The global utility of an assignment is then computed by aggregating local utilities according to a weighted averaging operator Δ, using the previously computed weights. The weights are values in [0,1] and their distribution is given by a decreasing depth in the graph: weight function defined over a specific interval (the lower the depth is, the higher the weight will be in the preference model), its outputs being subsequently normalized in order for them to sum to 1. Note that the weight distribution function might be a BUM (basic unit-interval monotonic) function [Yager, 2007]. A deeper study of this choice will be discussed in the future.

Such a weighted mean of local utilities works hand-in-hand with crisp local utilities. While local node utilities are computed as crisp values, the aggregated global utility, also crisp at first, can then be converted into a 2-tuple instance in order to offer a linguistic assessment while still preserving the precision of the original crisp value. If we switch to an end-to-end linguistic treatment, local utility values could be computed as linguistic terms, such as 2-tuples, using an *ad hoc* inference process, and then several linguistic aggregation operators [Xu, 2008], based on Yager's OWA operators [Yager, 1988], may be used to compute a linguistic global utility value. The choice of an appropriate aggregation operator proved to depend upon the application and shall be the point of a dedicated study. This choice could then be given to the user.

3.2.4 Service comparison and selection

The last step consists in comparing the different outcomes according to their global utility value to select one as the result of the overall decision process. Service comparison in order to make a binding choice between a consumer and producer are meant to be made automatically. In a fully automatic case using our LCP-net evaluation framework, crisp global utility values are used for comparison. But in other LCP-net application contexts, such a selection may rely on human intervention. In such a scenario, a fully linguistic approach, backed up by fuzzy 2-tuples, would take its full meaning, by providing the decision-maker with the qualitative assessment of linguistic terms but without loss of precision.

4 Case study

The following case study goes through the previously introduced steps focusing on a specific part of the imaging service preference model presented earlier in Figure 3.

4.1 Preference elicitation

In this model, security can be either *none* or *full*, given utilities *very low* and *very high* respectively. The preference of bandwidth over security is accounted for by an i-arc from B to S (an arc with a middle black triangle). The bandwidth is discretized using three linguistic variables B_L (low), B_M (medium) and B_H (high). Preferences among these values are given by the CPT beside B, expressing a *very low* preference for a low bandwidth, a *medium* one for the medium bandwidth and a *very high* one for a high bandwidth.

Image resolution is also discretized using two linguistic terms: R_L (low) and R_H (high). The preferences among these values are conditional to the bandwidth. If the bandwidth is low (B_L), a low resolution (R_L) has higher preference (*very high*), but if the bandwidth is high (B_H), a high resolution (R_H) is preferred (*very high*). When the bandwidth is medium (B_M), a low resolution image is preferred, but with less intensity (*high*).

The semantic of the linguistic terms used in the preference tables over bandwidth, resolution and utility is given beforehand by the fuzzy partitioning shown in Figure 4.

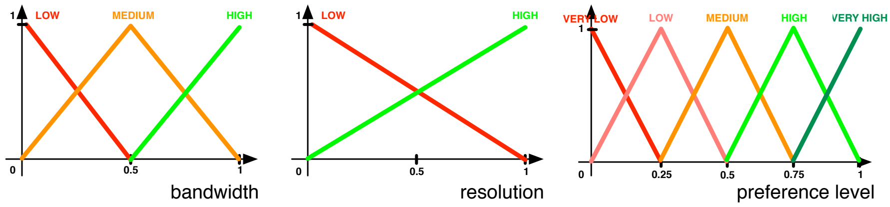

Figure 4: Fuzzy partitionings over bandwidth, resolution and utility.

After elicitation, the preference model previously shown in Figure 3 is fully obtained. In this particular model, bandwidth is always preferred over security, and if the bandwidth is low, low-resolution images are preferred because we need images as fast as possible.

4.2 Preference model translation

This section focuses uniquely on node R and its associated CPT, since the same process can be applied as-is to the other nodes. Given the CPT for node R, linked to node B in the preference model, and the previously mentioned fuzzy partitioning of bandwidth and resolution, we obtain after translation the following FIS specific to this node (see Figure 5) and defined here using the FCL language[4] used internally by our LCP-net evaluation framework.

As what has been previously said, input and output variables for this FIS (B, R and *Utility*) are declared as real (crisp value only). If a fuzzy value needs to be input, it first needs to be defuzzified, also output could afterward be translated to a linguistic representation.

4.3 Preference model evaluation

Still focusing on node R, the value of node B needs to be taken into account during evaluation. This is due to the fact that there is an arc between B and R in the preference model, and that the CPT attached to node R uses values taken by B as input.

During **QoS value injection** a bandwidth value of 30 kb/s is measured, then normalized over [0,1] to $B' = 0.30$, and finally fuzzified as a singleton, as seen in Figure 6.

[4] FCL stands for "Fuzzy Control Language", which is a standard for Fuzzy Control Programming published by the International Electrotechnical Commission [IEC, 2001].

```
FUNCTION_BLOCK fbName
VAR_INPUT
        B : REAL;
        R : REAL;
END_VAR
VAR_OUTPUT
        Utility : REAL;
END_VAR
FUZZIFY B
        TERM Bandwidth_High :=  (0.5, 0.0) (1.0, 1.0) ;
        TERM Bandwidth_Low :=  (0.0, 1.0) (0.5, 0.0) ;
        TERM Bandwidth_Medium :=  (0.0, 0.0) (0.5, 1.0) (1.0, 0.0) ;
END_FUZZIFY
FUZZIFY R
        TERM Resolution_High :=  (0.0, 0.0) (1.0, 1.0) ;
        TERM Resolution_Low :=  (0.0, 1.0) (1.0, 0.0) ;
END_FUZZIFY
DEFUZZIFY Utility
        TERM Utility_H :=  (0.5, 0.0) (0.75, 1.0) (1.0, 0.0) ;
        TERM Utility_L :=  (0.0, 0.0) (0.25, 1.0) (0.5, 0.0) ;
        TERM Utility_M :=  (0.25, 0.0) (0.5, 1.0) (0.75, 0.0) ;
        TERM Utility_VH :=  (0.75, 0.0) (1.0, 1.0) ;
        TERM Utility_VL :=  (0.0, 1.0) (0.25, 0.0) ;
        ACCU : MAX;
        METHOD : COG;
        DEFAULT := 0.0;
        RANGE := (0.0 .. 1.0);
END_DEFUZZIFY
RULEBLOCK Rules
        ACT : MIN;
        AND : MIN;
RULE 1 : IF (B is Bandwidth_Low) and (R is Resolution_Low) THEN Utility is Utility_VH;
RULE 2 : IF (B is Bandwidth_Low) and (R is Resolution_High) THEN Utility is Utility_VL;
RULE 3 : IF (B is Bandwidth_Medium) and (R is Resolution_Low) THEN Utility is Utility_H;
RULE 4 : IF (B is Bandwidth_Medium) and (R is Resolution_High) THEN Utility is Utility_L;
RULE 5 : IF (B is Bandwidth_High) and (R is Resolution_Low) THEN Utility is Utility_VL;
RULE 6 : IF (B is Bandwidth_High) and (R is Resolution_High) THEN Utility is Utility_VH;
END_RULEBLOCK
END_FUNCTION_BLOCK
```

Figure 5: FIS for node R CPT.

A fuzzy value for resolution is obtained and needs to be "adjusted" by a domain normalization on the [0,1] scale, as seen in Figure 7.

In any case, since B and R are declared in Figure 5 as real input variables, a fuzzy input will first need to be defuzzified before FIS evaluation.

Focusing on the **local utility value inference** for node B, only the previously normalized singleton bandwidth value is needed to compute its utility since it does not depend on any other node of the preference model. A visual output of the inference process for node B is given in Figure 8 using classical operators like Mamdani's fuzzy implication and Zadeh's T-Norm.

This figure shows that the fuzzy output of the inference process is defuzzified using a Center Of Gravity (COG) approach in order to obtain a local utility value for node B of 0.35 on the [0,1] utility domain. The same process will also be applied to node R and S in order to compute their respective local utilities.

In the **node weight computation** step, the weight distribution function used in this case study is given by the following formula defined over [1,100]:

$$g(x) = 1/x^2 + 0.8$$

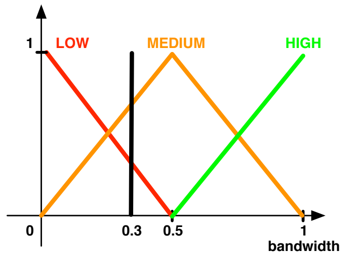

Figure 6: Singleton corresponding to measured bandwidth value.

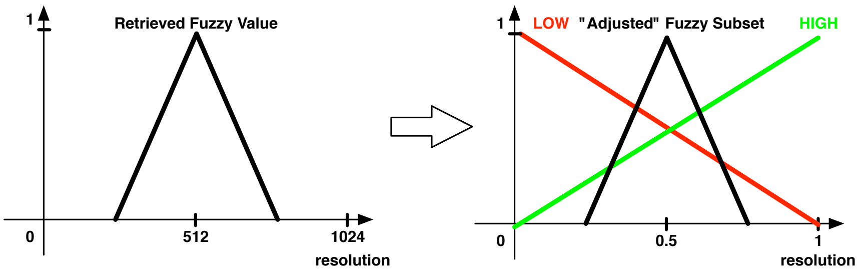

Figure 7: From measured fuzzy resolution value to adjusted fuzzy subset.

Given this function (see its overall shape in Figure 9), we obtain the following (*depth*, *weight*) pairs after normalization: (1, 0.529), (2, 0.235). These pairs are then associated to each node in the preference model as seen in Figure 10.

Finally, the process goes on with the **global utility value computation** for this preference model, given the following QoS value snapshot (B', S_{full}, R'), where B'=0.30, and the previously infered local utility values for each node: localUtility(B') = 0.35 (as was demonstrated earlier), localUtility(S_{full}) = 1 and localUtility(B',R') = 0.20.

The Δ operator is then applied according to the previously computed weights:

$$\Delta(B', S_{full}, R') = 0.529 \times 0.35 + 0.235 \times 1 + 0.235 \times 0.20 \approx 0.47$$

The global utility corresponding to the non-functional service offering (B', S_{full}, R') is approximately 0.47 on a scale from 0 to 1.

4.4 Case study summary

Figure 11 summarizes the steps undertaken by users for preference modeling before runtime and by our framework for setting-up service comparison and

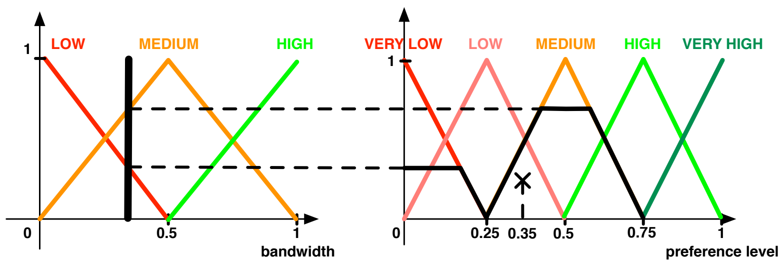

Figure 8: Fuzzy inference, from input bandwidth to utility output.

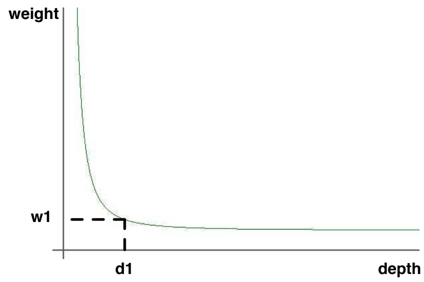

Figure 9: The increasing depth, weight function chosen for this case study.

selection at runtime. The final service selection step itself is not shown here as we focus on the global utility value computation of one particular service, with specific bandwidth and resolution QoS measured values.

4.5 Potential improvements

Based on the process summarized in Figure 11, some of the improvements expected from an end-to-end linguistic treatment discussed in the previous sections could be implemented as follows:

- the domain(s) fuzzy partitioning including the utility domain of Step 1 would provide for linguistic term sets of 2-tuples, i.e. pairs (s_i, α) with $\alpha = 0$. For example, the bandwidth would be represented as $\{$(*low*,0); (*medium*,0); (*high*,0)$\}$,
- in Step 2, the linguistic preference tables would also contain 2-tuples (e.g. (*very low*,0) or (*very low*,-0.2)) instead of simple linguistic terms (e.g. *very low*). Indeed the user expresses her preferences graphically, that is why a

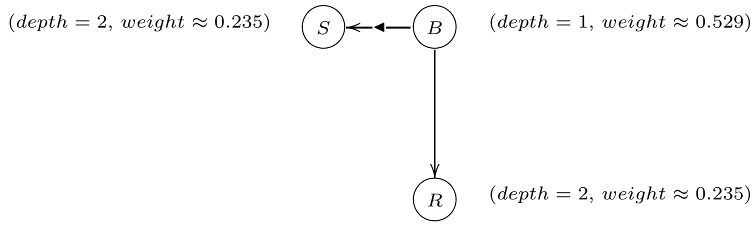

Figure 10: From node depths to node weights.

lateral displacement of the original linguistic term may appear. Thus the rules could be represented with 2-tuples: e.g. "If the resolution is (*low*,0) and the bandwidth is (*high*,0) then the utility is (*very low*,−0.2)",

- Step 3a would compile preferences through the fuzzy inference system (FIS) for 2-tuples, as proposed and discussed in [Alcalá et al., 2007],
- Step 4 would inject current QoS values converted into 2-tuples as well,
- meanwhile Step 3b wouldn't change,
- Step 5 would yield a local utility expressed by means of a 2-tuple,
- and Step 6 would aggregate weighted 2-tuples [Herrera and Martínez, 2000] in order to provide for a global utility expressed by means of a 2-tuple.

Note that using fuzzy 2-tuples shall also be very convenient in the case where the user afterwards would like to modify the domain variable partitioning: it shall be relevant to offer her the possibility to change the CPTs automatically (adding or deleting a column/line), i.e. the system would compute new utilities that could be expressed as (*very high*,−0.1)) for instance even if the user had first defined utilities only with $(s_i, 0)$ 2-tuples. Actually, as soon as we want to reconfigure the LCP-net automatically, we need to change the domain granularity and the 2-tuples appear a tool of choice for that, allowing the system to keep a relationship to the original user term set.

5 Conclusions

In this paper, we have adressed the problem of coping with the dynamism of Semantic Service-Oriented Architectures by proposing a new form of late-binding of services to calls in business processes based on the current values of candidate services QoS properties. This form of late-binding improves the capability of business processes to sustain their own QoS guarantees and implements a form of fault-tolerance by keeping a set of candidate services for each call as late as the execution of invocations at run-time.

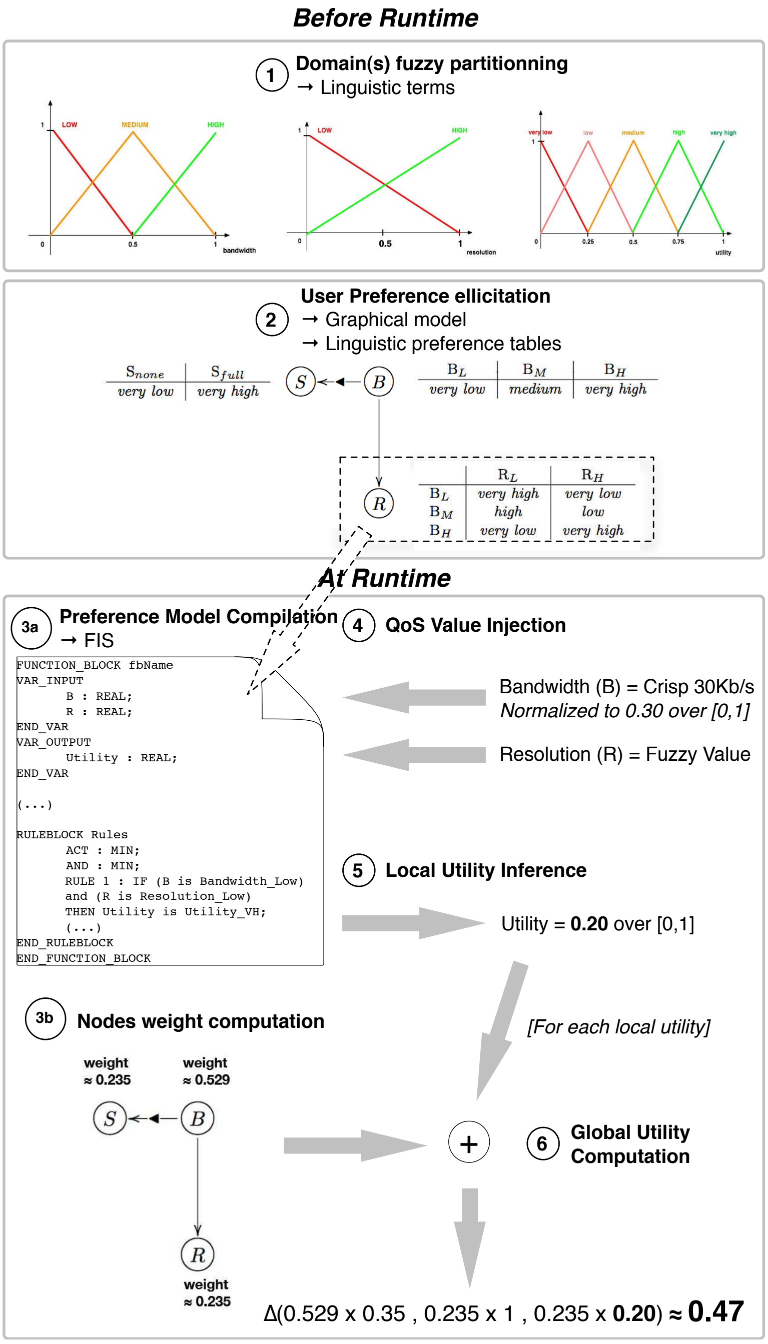

Figure 11: Case study summary.

To enable the selection of services in such a multi-criteria decision making process, preferences among their non-functional QoS properties must be expressed to get a total order among the candidates. To this end, the main contribution of this paper is to propose a new variant of CP-nets, called Linguistic CP-nets (LCP-nets), combining features of UCP-nets and TCP-nets with the advantages of a fuzzy linguistic approach to discretize continuous domain variables, and to express the utilities of assignments to variables in conditional preference tables. LCP-nets prove to be well suited for semantic SOA, as they allow users to effectively discretize continuous domain QoS with appropriate linguistic terms, and to express utility of assignments in a qualitative manner rather than an often contrived numerical one.

We have assumed for the time being that utilities are always expressed using the same linguistic term set, but this restriction could easily be removed in a fully linguistic approach based on 2-tuples by using a multigranular approach when computing the global utility function [Herrera et al., 2002]. Similarly, the assumption on the linguistic sets as being centered on 0.5 with terms being equidistant could also be removed using approaches to cope with unbalanced term sets [Herrera et al., 2001].

It shall be further stressed that although originally established for the specific SSOA context as target, the LCP-net formalism and its inference and computational framework are domain agnostic and could be easily applied to other similarly constrained contexts.

As future works, the next step is to implement the improvements discussed in section 4.5 but also to formalize the definition of LCP-nets as a variant of CP-nets. In particular, we shall prove that, in borderline cases — where there is no imprecision — the service ranking with fully linguistic LCP-nets is exactly the same than the ranking with *CP-nets but making preferences easier to express.

Acknowledgements

The research was partly funded by the French National Research Agency (ANR) *via* the SemEUsE project (ANR-07-TLOG-018). We would like to thank Thales Communications France for their constant support.

References

[Alcalá et al., 2007] Alcalá, R., Alcalá-Fdez, J., Herrera, F., and Otero, J. (2007). Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. *Int. J. Approx. Reasoning*, 44(1):45–64.

[Boubekeur and Tamine-Lechani, 2006] Boubekeur, F. and Tamine-Lechani, L. (2006). Recherche d'information flexible basée CP-Nets. In *Proc. Conference on Recherche d'Information et Applications (CORIA'06)*, pages 161–167.

[Boutilier et al., 2001] Boutilier, C., Bacchus, F., and Brafman, R. I. (2001). UCP-Networks: A directed graphical representation of conditional utilities. In *Proc. of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64.

[Boutilier et al., 2004] Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., and Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional *Ceteris Paribus* Preference Statements. *J. of Art. Intelligence Research*, 21:135–191.

[Brafman and Domshlak, 2002] Brafman, R. I. and Domshlak, C. (2002). Introducing variable importance tradeoffs into CP-nets. In *Proc. of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 69–76.

[Châtel et al., 2008] Châtel, P., Truck, I., and Malenfant, J. (2008). A linguistic approach for non-functional preferences in a semantic SOA environment. In *Computational Intelligence in Decision and Control, Proceedings of the 8th International FLINS Conference*, pages 889–894.

[Degani and Bortolan, 1988] Degani, R. and Bortolan, G. (1988). The Problem of Linguistic Approximation in Clinical Decision Making. *International J. of Approximate Reasoning*, 2:143–162.

[Delgado et al., 1993] Delgado, M., Verdegay, J., and Vila, M. (1993). On Aggregation Operations of Linguistic Labels. *International J. of Intelligent Systems*, 8:351–370.

[Driankov et al., 1993] Driankov, D., Hellendoorn, H., and Reinfrank, M. (1993). *An introduction to fuzzy control*. Springer-Verlag.

[Gonzales et al., 2008] Gonzales, C., Perny, P., and Queiroz, S. (2008). GAI-Networks: Optimization, Ranking and Collective Choice in Combinatorial Domains. *Foundations of computing and decision sciences*, 32(4):3–24.

[Herrera et al., 2001] Herrera, F., Herrera-Viedma, E., and Martínez, L. (2001). A Hierarchical Ordinal Model for Managing Unbalanced Linguistic Term Sets Based on the Linguistic 2-Tuple Model. In *EUROFUSE Workshop on Preference Modelling and Applications*, pages 201–206.

[Herrera and Martínez, 2000] Herrera, F. and Martínez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems*, 8(6):746–752.

[Herrera et al., 2002] Herrera, F., Martínez, L., Herrera-Viedma, E., and Chiclana, F. (2002). Fusion of Multigranular Linguistic Information based on the 2-tuple Fuzzy Linguistic Representation Model. In *Proceedings of IPMU 2002*, pages 1155–1162.

[IEC, 2001] IEC (2001). IEC 61131-7 Fuzzy Control Programming.

[Lausen and Innsbruck, 2007] Lausen, H. and Innsbruck, D. (2007). *Semantic Annotations for WSDL and XML Schema*.

[Schröpfer et al., 2007] Schröpfer, C., Binshtok, M., Shimony, S. E., Dayan, A., Brafman, R., Offermann, P., and Holschke, O. (2007). Introducing preferences over NFPs into service selection in SOA. In *Proc. Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'07)*.

[Truck and Akdag, 2006] Truck, I. and Akdag, H. (2006). Manipulation of Qualitative Degrees to Handle Uncertainty : Formal Methods and Applications. *Knowledge and Information Systems (KAIS)*, 9(4):385–411.

[Xu, 2008] Xu, Z. (2008). *Linguistic Aggregation Operators: An Overview*, volume 220, pages 163–181. Springer Verlag. ISBN: 978-3-540-73722-3.

[Yager, 1988] Yager, R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190.

[Yager, 2007] Yager, R. (2007). Using Stress Functions to Obtain OWA Operators. *IEEE Trans. on Fuzzy Systems*, 15(6):1122–1129.

[Zadeh, 1975] Zadeh, L. (1975). The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. *Information Sciences, Part I, II, III*, 8,8,9:199–249, 301–357, 43–80.