Fuzzy Description Logics for Bilateral Matchmaking in e-Marketplaces

Azzurra Ragone[1,4], Umberto Straccia[2], Fernando Bobillo[3], Tommaso Di Noia[4], Eugenio Di Sciascio[4]

[1] University of Michigan, Ann Arbor, MI, USA – aragone@umich.edu
[2] ISTI-CNR, Pisa, Italy – straccia@isti.cnr.it
[3] Department of Computer Science and Artificial Intelligence, University of Granada – fbobillo@decsai.ugr.es
[4] SisInfLab, Politecnico di Bari, Bari, Italy – {t.dinoia,disciascio}@poliba.it

Abstract. We present a novel Fuzzy Description Logic (DL) based approach to automate matchmaking in e-marketplaces. We model traders' preferences with the aid of Fuzzy DLs and, given a request, use utility values computed w.r.t. Pareto agreements to rank a set of offers. In particular, we introduce an expressive Fuzzy DL, extended with concrete domains in order to handle numerical, as well as non numerical features, and to deal with vagueness in buyer/seller preferences. Hence, agents can express preferences as *e.g.*, *I am searching for a passenger car costing about 22000€ yet if the car has a GPS system and more than two-year warranty I can spend up to 25000€*. We note that, among all possible matches, our matchmaking approach chooses the mutually beneficial ones.

1 Introduction

In an e-marketplace, a transaction can be organized in three different stages [25]: *discovery*, *negotiation* and *execution*. During the discovery phase, the marketplace helps the buyer to look for promising offers best matching her request. The result of this **matchmaking** phase is a ranked list of offers (usually ranked with respect to buyer's preferences). In the eventual *negotiation* phase, the marketplace guides the buyer and the seller to reach an agreement. With the *execution* of the transaction, the buyer and the seller exchange the good.

Usually negotiation and matchmaking are two distinct processes executed sequentially. First, the marketplace ranks offers for the buyer taking into account her request, *i.e.*, her preferences expressed w.r.t. some utility function, then, usually, a negotiation starts with the seller having the best ranked supply, in order to reach an agreement that satisfies both traders. That is, the marketplace tries to find an agreement which is Pareto efficient [5], as well as beneficial for both traders [16].

Looking at the nature of matchmaking and negotiation we see that in the former there is only one active actor – the buyer – while in the latter we have two active actors – both traders. A typical marketplace uses only buyer's preferences for discovery and both traders' preferences for negotiation. In a few words we can say that discovery is

[5] An agreement is Pareto efficient when it is not possible to improve the utility of one trader, without lowered the utility of the opponent one.

"unilateral", while negotiation is "bilateral". Due to this difference, it might occur often that an offer resulting promising for the buyer *i.e.*, with a good satisfaction degree for her preferences, does not lead to an agreement because, on the other side, seller's preferences are not adequately satisfied.

The idea behind the approach we propose in this paper is to merge the discovery and negotiation phase in a **bilateral matchmaking**. In our bilateral matchmaking scenario given a buyer's request and a set of supplies, the matchmaker computes for each supply a Pareto-efficient agreement maximazing the degree of satisfaction of the traders (see Section 4), and then ranks all these agreements w.r.t. the utility of the buyer.

We set our framework in an e-marketplace selling highly differentiated products (*e.g.*, cars, Personal Computers, travel services, etc.), therefore there is a need for a Knowledge Representation language able to model not only complex user preferences on set of issues, but also relations among the issues themselves.

We propose here a *fuzzy Description Logic* (see, [24] for an overview) endowed with *concrete domains* to model relations among issues and as a communication language between traders. We may represent facts such as that a *Ferrari is an Italian car maker* ($Ferrari \sqsubseteq ItalianMaker$), or that a *Sedan is a type of Passenger Car* ($Sedan \sqsubseteq PassengerCar$), or the fact that a car cannot have at the same time a fuel that is both *Diesel and GAS* ($Diesel \sqcap Gasoline \sqsubseteq \bot$). Such kind of relations can be expressed in a Theory (from now on an Ontology) $\mathcal{T}$. Furthermore, we may represent *preferences*, such as *e.g.*, a seller can state that *"If you want an embedded alarm system you'll have to wait more than one month"* ($AlarmSystem \sqsubseteq (\geqslant deliverytime\ 30)$), as well as a buyer can state that *"I would like a passenger car with an alarm system if it costs more than 25000€"*($PassengerCar \sqcap (\geqslant price\ 25000)) \sqsubseteq AlarmSystem$)). In our proposal, concrete domains allow to deal with numerical features, which are mixed, in preferences, with non numerical ones.

We note that in this scenario a buyer request, as well as a seller supply, can be split into two parts: one involving issues that have to be necessarily satisfied in order to accept a final agreement, which we call *hard constraints*, and another one involving issues buyer and seller are willing to negotiate on, we call these *soft constraints*. Among *soft constraints* there can be also *fuzzy constraints*, which are preferences involving numerical features. *Fuzzy constraints* are represented in our approach using fuzzy membership functions, see Section 2, therefore while a simple *soft constraint* can or cannot be satisfied, a *fuzzy constraints* can also be satisfied to a "certain degree". For example, a buyer can state, among soft constraints, that *if a GPS system is mounted on the car she can spend up to 25000 for a sedan*; if the price in the proposed agreement is equal to 25500 we should not simply say that the preference is not satisfied at all, but rather that is satisfied to a certain degree, as will be better described later on (see Section 3).

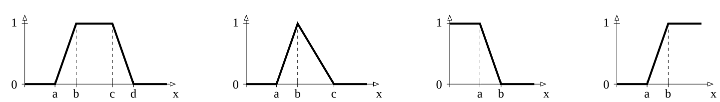
We note that in our framework it is possible to model *positive* and *negative* preferences (I would like a car black or gray, but not red), as well as *conditional preferences* (I would like leather seats if the car is black) involving both numerical features and non numerical ones (If you want a car with GPS system you have to wait at least one month) or only numerical ones (I accept to pay more than 25000€ only if there is more than a two-year warranty).

Besides we model *quantitative* preferences; thanks to the weight assigned to each preference it is possible to determine a relative importance among them, rather than only a total order between them. Obviously, the whole approach holds also if the user does not specify a weight for each preference, but only a global order on preferences. However, in that case, the relative importance among preferences is missed.

The rest of the paper is structured as follows: next section discusses the fuzzy language we adopt in order to express traders' preferences. In Section 3 we set the stage of the the bilateral matchmaking problem in fuzzy DL and then we illustrate how to compute Pareto agreements. In Section 5 the whole process is highlighted with the aid of a simple example. Related Work and discussion close the paper.

2 A Fuzzy DL to express preferences

In a bilateral matchmaking scenario traders express preferences involving numerical as well as non numerical issues, in some way interrelated. The variables representing numerical features are either involved in *hard constraints* or *soft constraints*. In hard constraints, the variables are always constrained by comparing them to some constant, like $(\leqslant \mathit{price}\ 20.000)$, or $(\geqslant \mathit{month\_warranty}\ 60)$, and such constraints can be combined into complex requirements, *e.g.*, $\mathit{Sedan} \sqcap (\leqslant \mathit{price}\ 25.000) \sqcap (\leqslant \mathit{deliverytime}\ 30)$ (representing a sedan, costing no more than 25.000 euros, delivered in at most 30 days), or $\mathit{AlarmSystem} \sqcap (\geqslant \mathit{price}\ 26.000)$ (expressing the seller's requirement "if you want an alarm system mounted you'll have to spend at least 26.000 euros"). Vice-versa when numerical features are involved in *soft constraints*, also called *fuzzy constraints*, the variables representing numerical features are constrained by so-called fuzzy membership functions, as shown below.



$trz(a, b, c, d)$ $tri(a, b, c)$ $ls(a, b)$ $rs(a, b)$

For instance, $(\exists price.ls(18000, 22000))$ dictates that given a price it returns the degree of truth to which the constraint is satisfied. Essentially, $(\exists price.ls(18000, 22000))$ states that if the price is no higher than 18000 then the constraint is definitely satisfied, while if the price is higher than 22000 then the constraint is definitely not satisfied. In between 18000 and 22000, we use linear interpolation, given a price, to evaluate the satisfaction degree of the constraint.

Fuzzy DL syntax. Now, we specify the syntax of our fuzzy DL for matchmaking. The fuzzy DL considers the salient features of the fuzzyDL reasoner *fuzzyDL* [6] (see [3]). The basic fuzzy DL we consider is the fuzzy DL $\mathcal{SHIF}(D)$ [24], *i.e.*, $\mathcal{SHIF}$ with concrete data types. But, for our purpose, we do not need individuals and assertions. So, let us consider an alphabet for *concepts names* (denoted A), *abstract roles names* (denoted R), *i.e.*, binary predicates *concrete roles names* (denoted T), and *modifiers* (denoted m). $\mathbf{R}_a$ also contains a non-empty subset $\mathbf{F}_a$ of *abstract feature names* (denoted r), while $\mathbf{R}_c$ contains a non-empty subset $\mathbf{F}_c$ of *concrete feature names* (denoted t). Features are functional roles. Concepts in fuzzy $\mathcal{SHIF}$ (denoted C, D) are build as usual from atomic concepts A and roles R: $\top, \bot, A, C \sqcap D, C \sqcup D, \neg C, \forall R.C$ and $\exists R.C$. Now,

[6] http://gaia.isti.cnr.it/ straccia/software/fuzzyDL/fuzzyDL.html

Fuzzy $\mathcal{SHIF}(D)$ extends $\mathcal{SHIF}$ with concrete data types [1], *i.e.*, it has the additional concept constructs $\forall T.d$, $\exists T.d$ and DR, where

$$\begin{array}{l} d \;\; \rightarrow ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid trz(a,b,c,d) \\ DR \rightarrow (\geqslant t\ val) \mid (\leqslant t\ val) \mid (= t\ val) \end{array}$$

and val is an integer or a real depending on the range of the concrete feature t. For instance, the expression $Sedan \sqcap (\leqslant price\ 25.000)$ will denote the set of sedans costing no more than 25.000 euros, while $Sedan \sqcap (\exists price.ls(18000, 22000))$, says informally, specifies the class of sedans with a price whose degree of satisfaction is determined by $ls(18000, 22000)$. Finally, we further extend $\mathcal{SHIF}(D)$ as follows:

$$C, D \rightarrow (w_1C_1 + w_2C_2 + \ldots + w_kC_k) \mid C[\geqslant n] \mid C[\leqslant n]$$

where $n \in [0,1]$, $w_i \in [0,1]$, $\sum_{i=1}^{k} w_i = 1$. The expression $(w_1C_1+w_2C_2+\ldots+w_kC_k)$ denotes a weighted sum, while $C[\geqslant n]$ and $C[\leqslant n]$ are threshold concepts.

A *fuzzy DL ontology* (also Knowledge Base, KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{R} \rangle$ consists of a fuzzy TBox $\mathcal{T}$ and a fuzzy RBox $\mathcal{R}$. A *fuzzy TBox* $\mathcal{T}$ is a finite set of *fuzzy General Concept Inclusion axioms* (GCIs) $\langle C \sqsubseteq D, n \rangle$, where $n \in (0,1]$ and C, D are concepts. If the truth value n is omitted then the value 1 is assumed. Informally, $\langle C \sqsubseteq D, n \rangle$ states that all instances of concept C are instances of concept D to degree n, that is, the subsumption degree between C and D is at least n. For instance, $\langle Sedan \sqsubseteq PassengerCar, 1 \rangle$ states that a sedan is a passenger car. We write $C = D$ as a shorthand of the two axioms $\langle C \sqsubseteq D, 1 \rangle$ and $\langle D \sqsubseteq C, 1 \rangle$. Axioms of the form $A = D$ are called *concept definitions* (*e.g.*, , $InsurancePlus = DriverInsurance \sqcap TheftInsurance$). A *fuzzy RBox* $\mathcal{R}$ is a finite set of role axioms of the form: (i) $(fun\ R)$, stating that a role R is functional, *i.e.*, R is a feature; (ii) $(trans\ R)$, stating that a role R is transitive; (iii) $R_1 \sqsubseteq R_2$, meaning that role R_2 subsumes role R_1; and (iii) $(inv\ R_1\ R_2)$, stating that role R_2 is the inverse of R_1 (and vice versa). A simple role is a role which is neither transitive nor has a transitive subroles. An important restriction is that functional needs to be simple.

Fuzzy DL semantics [3]. The main idea is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain. Therefore, axioms, rather than being "classical" evaluated (being either true or false), they are "many-valued" evaluated, *i.e.*, their evaluation takes a degree of truth in $[0,1]$.

A *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a concrete domain $D = \langle \Delta_D, C(D) \rangle$ consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*), disjoint from Δ_D, and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns: (i) to each abstract concept C a function $C^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \rightarrow [0,1]$; (ii) to each abstract role R a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0,1]$; (iii) to each abstract feature r a partial function $r^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0,1]$ such that for all $x \in \Delta^{\mathcal{I}}$ there is an unique $y \in \Delta^{\mathcal{I}}$ on which $r^{\mathcal{I}}(x,y)$ is defined; (iv) to each concrete role T a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_D \rightarrow [0,1]$; (v) to each concrete feature t a partial function $t^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_D \rightarrow [0,1]$ such that for all $x \in \Delta^{\mathcal{I}}$ there is an unique $v \in \Delta_D$ on which $t^{\mathcal{I}}(x,v)$ is defined. In order to extend the mapping, the interpretation function $\cdot^{\mathcal{I}}$ is extended to roles and complex concepts, we need functions to define the negation, conjunction, disjunction (called norms), etc of values in $[0,1]$. The choice of them is not arbitrary. Some well-known specific choices are described in the table below.

	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh
$\ominus x$	$1 - x$	if $x = 0$ then 1 else 0	if $x = 0$ then 1 else 0	$1 - x$
$x \otimes y$	$\max(x + y - 1, 0)$	$\min(x, y)$	$x \cdot y$	$\min(x, y)$
$x \oplus y$	$\min(x + y, 1)$	$\max(x, y)$	$x + y - x \cdot y$	$\max(x, y)$
$x \Rightarrow y$	if $x \leqslant y$ then 1 else $1 - x + y$	if $x \leqslant y$ then 1 else y	if $x \leqslant y$ then 1 else y/x	$\max(1 - x, y)$

The next table highlights some salient properties of them.

Property	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh Logic
$x \otimes \ominus x = 0$	•	•	•	
$x \oplus \ominus x = 1$	•			
$x \otimes x = x$		•		•
$x \oplus x = x$		•		•
$\ominus \ominus x = x$	•			•
$x \Rightarrow y = \ominus x \oplus y$	•			•
$\ominus (x \Rightarrow y) = x \wedge \neg y$	•			•
$\ominus (x \otimes y) = \ominus x \oplus \ominus y$	•	•	•	•
$\ominus (x \oplus y) = \ominus x \otimes \ominus y$	•	•	•	•

It is important to note that we can never enforce that a choice of the interpretation of the connectors satisfies all listed properties, because then the logic will collapse to classical boolean propositional logic.

Now, the mapping $\cdot^{\mathcal{I}}$ is extended to roles, complex concepts and GCIs as follows:

$$
\begin{array}{rclrcl}
\bot^{\mathcal{I}}(x) & = & 0 & (= t\ val)^{\mathcal{I}}(x) & = & \sup_{c \in \Delta_D} t(x, v) \otimes (v = val) \\
\top^{\mathcal{I}}(x) & = & 1 & (\forall R.C)^{\mathcal{I}}(x) & = & \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y) \\
(\neg C)^{\mathcal{I}}(x) & = & \ominus C^{\mathcal{I}}(x)) & (\exists R.C)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y) \\
(C \sqcap D)^{\mathcal{I}}(x) & = & C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x) & (\forall T.d)^{\mathcal{I}}(x) & = & \inf_{y \in \Delta_D} T^{\mathcal{I}}(x, v) \Rightarrow d^{\mathcal{I}}(y) \\
(C \sqcup D)^{\mathcal{I}}(x) & = & C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x) & (\exists T.d)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta_D} T^{\mathcal{I}}(x, v) \otimes d^{\mathcal{I}}(y) \\
(\geqslant t\ val)^{\mathcal{I}}(x) & = & \sup_{c \in \Delta_D} t(x, v) \otimes (v \geqslant val) & (\leqslant t\ val)^{\mathcal{I}}(x) & = & \sup_{c \in \Delta_D} t(x, v) \otimes (v \leqslant val)
\end{array}
$$

$$((w_1 C_1 + w_2 C_2 + \ldots + w_k C_k)^{\mathcal{I}}(x) = w_1 {C_1}^{\mathcal{I}}(x) + \cdots + w_k {C_k}^{\mathcal{I}}(x)$$

$$(C[\geqslant\ n])^{\mathcal{I}}(x) = \begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \geqslant n \\ 0, & \text{otherwise} \end{cases} \qquad (C[\leqslant\ n])^{\mathcal{I}}(x) = \begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \leqslant n \\ 0, & \text{otherwise} \end{cases}$$

$$(C \sqsubseteq D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$$

The notion of *satisfaction* of a fuzzy axiom E by a fuzzy interpretation $\mathcal{I}$, denoted $\mathcal{I} \models E$, is defined as follows: $\mathcal{I} \models \langle \tau \geqslant n \rangle$ iff $\tau^{\mathcal{I}} \geqslant n$, $\mathcal{I} \models (trans\ R)$ iff $\forall_{x,y \in \Delta^{\mathcal{I}}}, R^{\mathcal{I}}(x, y) \geqslant \sup_{z \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, z) \otimes R^{\mathcal{I}}(z, y)$, $\mathcal{I} \models R_1 \sqsubseteq R_2$ iff $\forall x, y \in \Delta^{\mathcal{I}}. {R_1}^{\mathcal{I}}(x, y) \leqslant {R_2}^{\mathcal{I}}(x, y)$, and $\mathcal{I} \models (inv\ R_1 R_2)$ iff $\forall x, y \in \Delta^{\mathcal{I}}. {R_1}^{\mathcal{I}}(x, y) = {R_2}^{\mathcal{I}}(y, x)$.

For a set of axioms $\mathcal{E}$, we say that $\mathcal{I}$ *satisfies* $\mathcal{E}$ iff I satisfies each element in $\mathcal{E}$. We say that $\mathcal{I}$ is a *model* of E (resp. $\mathcal{E}$) iff $\mathcal{I} \models E$ (resp. $\mathcal{I} \models \mathcal{E}$). $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, denoted $\mathcal{I} \models \mathcal{K}$, iff $\mathcal{I}$ is a model of each component $\mathcal{A}, \mathcal{T}$ and $\mathcal{R}$, respectively. An axiom E is a *logical consequence* of a knowledge base $\mathcal{K}$, denoted $\mathcal{K} \models E$ iff every model of $\mathcal{K}$ satisfies E. Given $\mathcal{K}$, the *best satisfiability bound* of a concept C, denoted $bsb(\mathcal{K}, C)$, is

$$bsb(\mathcal{K}, C) = \sup_{\mathcal{I}} \sup_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \mid \mathcal{I} \models \mathcal{K}\}\ .$$

Essentially, among all models $\mathcal{I}$ of the KB, we are determining the maximal degree of truth that the concept C may have over all individuals $x \in \Delta^{\mathcal{I}}$.

We conclude this section with the following remark. *For the sake of our purpose, for the remainder of the paper we will use Łukasiewicz Logic as the specific interpretation of the connectives.* The reason for this choice is due to the nice logical and computational properties of Łukasiewicz Logic. Furthermore, if we use $\cdot_G$ to represent Gödel operators, we note that $x \otimes_G y = \min(x, y)$ and $x \oplus_G y = \max(x, y)$ can also be defined in it by means of $x \otimes (x \rightarrow y)$ and $\ominus(\ominus x \otimes_G \ominus y)$, respectively. As a consequence, we may define the following additional macros on concepts: let C, D be concepts, then $C \rightarrow D := \neg C \sqcup D$, $C \sqcap_G D := C \sqcap (C \rightarrow D)$, and $C \sqcup_G D := \neg(\neg C \sqcap_G D))$ are concepts as well.

We know that a solution maximizing the sum of traders' preference value is Pareto optimal then, intuitively, a further important property for our purpose is that in Łukasiewicz Logic the conjunction function allows us to determine Pareto optimal solutions in the following sense.

Proposition 1. *If the maxima of $x \otimes_Ł y$, with $\langle x, y\rangle \in S \subseteq [0, 1] \times [0, 1]$, where $\otimes_Ł$ is Łukasiewicz t-norm, is positive then the maxima is also Pareto optimal.*

As we will see later on, relying on Łukasiewicz logic will guarantee that the solutions of the bilateral matchmaking process are then Pareto optimal ones. Note also that the maxima of $x \otimes_G y$, with $\langle x, y\rangle \in S$, is not Pareto optimal.

3 Multi Issue Bilateral Matchmaking in fuzzy DLs

Marketplaces are typical scenarios where the notion of fuzziness appears frequently. The concept of Cheap or Expensive are quite usual. In a similar way it is common to have a fuzzy interpretation of numerical constraints. If a buyer looks for a car with a price lesser than 15,000 €and a supplier selling his car for 15,500 €, we can not say they do not match at all. Actually, they match with a certain degree. Hence, a fuzzy language, as the one we presented in the previous sections, would be very useful to model demands and supplies in matchmaking scenarios.

Similarly to the approach proposed in [19], we propose to use our fuzzy DL to represent both buyer's demand and seller's supply and represent relations among issues, both abstract and numerical, by a fuzzy DL knowledge base.

As introduced in Section 1, in bilateral matchmaking scenarios, both buyer's request and seller's offer can be split into *hard constraints* and *soft constraints*. *Hard constraints* represent what has to be (necessarily) satisfied in the final agreement; *soft constraints* represent traders' preferences.

Example 1. Consider the example where buyer's request is: "I am searching for a Passenger Car equipped with Diesel engine. I need the car as soon as possible, and I can not wait more than one month. Preferably I would like to pay less than 22,000 € furthermore I am willing to pay up to 24,000 € if warranty is greater than 160000 km. I won't pay more than 27,000 €'.

Here, it is easy to see the difference between hard constraints and soft ones:

hard constraints : I want a Passenger Car provided with a Diesel engine. I can not wait more than one month. I won't pay more than 27,000 € .

soft constraints : I would like to pay less than 22,000 € furthermore I am willing to pay up to 24,000 € if warranty is greater than 160000 km.

Definition 1 (Demand, Supply, Agreement). *Given an ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}\rangle$ representing the knowledge on a marketplace domain*

– *a* demand *is a concept definition* β *of the form* $B = C[\geqslant 1.0]$ *(for Buyer) such that* $\langle \mathcal{T} \cup \{\beta\}, \mathcal{R}\rangle$ *is satisfiable.*
– *a seller's* supply *is a concept definition* σ $S = D[\geqslant 1.0]$ *(for Seller) such that* $\langle \mathcal{T} \cup \{\sigma\}, \mathcal{R}\rangle$ *is satisfiable.*
– $\mathcal{I}$ *is a* possible deal *between* β *and* σ *iff* $\mathcal{I} \models \langle \mathcal{T} \cup \{\sigma, \beta\}, \mathcal{R}\rangle$*. We also call* $\mathcal{I}$ *an* agreement*.*

σ and β represent the minimal requirements needed in the final agreement. As they are mandatory the threshold value is set to 1.0, meaning that they have to be in the agreement. Obviously, if seller and buyer have set *hard constaints* that are in conflict with each other, that is $\langle \mathcal{T} \cup \{\beta, \sigma\}, \mathcal{R}\rangle$ has no models, then it is impossible to reach an agreement, *i.e.*, the set of possible deals is empty. If the buyer is interested in a conflicting supply it is necessary a revision of her *hard constaints*.

In the bilateral matchmaking process, besides hard constraints, both traders can express preferences on some (bundle of) issues. In our fuzzy DL framework preferences can be represented as weighted formulae (see Section 2). More formally:

Definition 2 (Preferences). *The buyer's* preference $\mathcal{B}$ *is a weighted concept of the form* $n_1 \cdot \beta_1 \sqcup \ldots \sqcup n_k \cdot \beta_k$*, where each* β_i *represents the subject of a buyer's preference, and* n_i *is the weight associated to it. Analogously, the seller's* preference $\mathcal{S}$ *is a weighted concept of the form* $m_1 \cdot \sigma_1 \sqcup \ldots \sqcup m_h \cdot \sigma_h$*, where each* σ_i *represents the subject of a seller's preference, and* m_i *is the weight associated to it.*

For instance, the Buyer's request in Example 1 is formalized as:

$$\beta \text{ is } B = (PassengerCar \sqcap Diesel \sqcap (price \leqslant 27,000) \sqcap (deliverytime \leqslant 30))[\geqslant 1.0]$$
$$\beta_1 = (\exists price.ls(\mathit{22000}, \mathit{25000}))$$
$$\beta_2 = (\exists km\_warranty.rs(\mathit{140000}, \mathit{160000})) \rightarrow (\exists price\ , ls(\mathit{24000}, \mathit{27000}))$$

where $price$ and $km\_warranty$ are concrete features. We normalize the sum of the weights of both agents' to 1 to eliminate outliers, and make the set of preferences comparable.

The utility function, that we call ***preference utility***, is then a weighted sum of the preferences satisfied in the agreement.

Dealing with concrete features, we always have to set a ***reservation value*** [21] represented as a *hard constraint*. It is the maximum (or minimum) value in the range of possible feature values to reach an agreement. If we consider Example 1 we see that the buyer expresses two reservation values, one on price *"more than 27,000 €"* and the other on delivery time *"less than 1 month"*.

Reservation value is the maximum (or minimum) value in the range of possible feature values to reach an agreement, *e.g.*, the maximum price the buyer wants to pay for a car or the minimum warranty required, as well as, from the seller's perspective the minimum price he will accept to sell the car or the minimum delivery time. Usually, each participant knows its own reservation value and ignores the opponent's one[7]. In the following, given a concrete feature f we refer to reservation values of buyer and seller on f with $r_{\beta,f}$ and $r_{\sigma,f}$ respectively.

[7] Actually, it is possible that traders know probability distributions of opponent's reservation value.

Since *reservation values* represent *hard constraints* then buyer's ones are added to β and seller's ones to σ (see Example 1).

The last elements we have to introduced in order to formally define an agreement in a bilateral matchmaking process are *disagreement thresholds*, also called disagreement payoffs, t_β, t_σ. They represent the minimum utility that the agent need to reach to accept the agreement. Minimum utilities may incorporate an agent's attitude toward concluding the transaction, but also overhead costs involved in the transaction itself, *e.g.*, fixed taxes.

Definition 3. *Given an ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{R} \rangle$, a demand β, a set of buyer's preferences $\mathcal{B}$ and a disagreement threshold t_β, a supply σ and a set of seller's preferences $\mathcal{S}$ and a disagreement threshold t_σ, an agreement in a bilateral matchmaking process is a model $\mathcal{I}$ of*

$$\bar{\mathcal{K}} = \langle \mathcal{T} \cup \{\sigma, \beta\} \cup \{Buy = (\mathcal{B}[\geqslant t_\beta]), Sell = (\mathcal{S}[\geqslant t_\sigma])\}, \mathcal{R} \rangle \ .$$

Clearly, not every agreement $\mathcal{I}$ is beneficial both for the buyer and for the seller. We need a criterion to find the optimal mutual agreement. Given a demand and a set of supplies, for each of them we will compute the optimal agreement with the demand and we will rank them with respect to the buyer's utility value in the optimal agreement itself.

4 Computing Pareto agreements

To compute an optimal agreement we rely on the notion of Pareto agreement. Given an ontology $\mathcal{K}$ representing a set of constraints, we are interested in agreements that are Pareto-efficient, in order to make traders as much as possible satisfied. In our fuzzy DL based framework, in order to compute a *Pareto agreement* we procede as follows.

Let $\mathcal{K}$ be a fuzzy DL ontology, let β be the buyer's demand, let σ be the seller's supply, let $\mathcal{B}$ and $\mathcal{S}$ be respectively the buyer's and seller's preferences. We define $\bar{\mathcal{K}}$ as the ontology

$$\bar{\mathcal{K}} = \langle \mathcal{T} \cup \{\sigma, \beta\} \cup \{Buy = (\mathcal{B}[\geqslant t_\beta]), Sell = (\mathcal{S}[\geqslant t_\sigma])\}, \mathcal{R} \rangle \ .$$

In $\bar{\mathcal{K}}$, the concept Buy collects all the buyer's preferences. Hence, the higher is the maximal degree of satisfiability of Buy (*i.e.*, $bsb(\bar{\mathcal{K}}, Buy)$), the more the buyer is satisfied. Similarly, the concept $Sell$ collects all the seller's preferences in such a way that the higher is the maximal degree of satisfiability of $Sell$ (*i.e.*, $bsb(\bar{\mathcal{K}}, Sell)$), the more the seller is satisfied. Now, it is clear that the best agreement among the buyer and the seller is the one assigning the maximal degree of satisfiability to the conjunction $Buy \sqcap Sell$ (remember we use Łukasiewicz semantics). In formulae, once we determine

$$v_P = bsb(\bar{\mathcal{K}}, Buy \sqcap Sell) \ ,$$

we can say that a *Pareto agreement* is a model $\bar{\mathcal{I}}$ of $\bar{\mathcal{K}}$ such that

$$v_P = \sup_{x \in \Delta^{\mathcal{I}}} (Buy \sqcap Sell)^{\mathcal{I}}(x) > 0 \ ,$$

that is the *Pareto agreement value* is attained at $\bar{\mathcal{I}}$ and has to be positive.

A Pareto agreement can be computed using the *fuzzyDL* reasoner.

5 The matchmaking process

Summing up, given a demand and a set of supplies, the bilateral matchmaking process is executed covering the following steps:

Initial Setting. The buyer defines *hard constraints* β and preferences (*soft constraints*) $\mathcal{B}$ with corresponding weights for each preference $n_1, n_2, ..., n_k$, as well as the threshold t_β. The same did the sellers when they posted the description of their supply within the marketplace[8]. Notice that for numerical features involved in the negotiation process, both in β and σ their respective reservation values are set either in the form $(\leqslant f\ r_f)$ or in the form $(\geqslant f\ r_f)$.

Find Optimal Agreements. For each supply in the marketplace, the matchmaker computes the corresponding Pareto agreement (see Section 4). Without loss of generality, here we assume there exists only one ontology. In case more than one ontology exist, before finding Pareto agreements the matchmaker has to perform an ontology matching process in order to make all supplies comparable with the demand.

Ranking. Given a supply σ_i and the corresponding optimal agreement $\bar{\mathcal{I}}_i$, we rank σ_i w.r.t. the value of $Buy^{\bar{\mathcal{I}}_i}$, *i.e.*, w.r.t. the buyer's degree of satisfiability.

Let us present a tiny example in order to better clarify the approach. For the sake of simplicity, we will consider only one seller, clearly, in a real case scenario, the whole process will be repeated for each seller's supply posted in the e-marketplace. Given the toy ontology $\mathcal{K} = \langle \mathcal{T}, \emptyset \rangle$, with

$$\mathcal{T} = \begin{cases} \mathit{Sedan} \sqsubseteq \mathit{PassengerCar} \\ \mathit{ExternalColorBlack} \sqsubseteq \neg \mathit{ExternalColorGray} \\ \mathit{SatelliteAlarm} \sqsubseteq \mathit{AlarmSystem} \\ \mathit{InsurancePlus} = \mathit{DriverInsurance} \sqcap \mathit{TheftInsurance} \\ \mathit{NavigatorPack} = \mathit{SatelliteAlarm} \sqcap \mathit{GPS\_system} \end{cases}$$

The buyer and the seller specify their *hard* and *soft constraints*. For each numerical feature involved in *soft constraints* we associate a fuzzy function. If the bargainer has stated a reservation value on that feature, it will be used in the definition of the fuzzy function, otherwise a default value will be used.

β is $B = \mathit{PassengerCar} \sqcap (\leqslant \mathit{price}\ 26000)[\geqslant 1.0]$
$\beta_1 = ((\exists HasAlarmSystem.AlarmSystem) \rightarrow (\exists Has\ Price.L(22300, 22750)))$
$\beta_2 = ((\exists HasInsurance.DriverInsurance) \sqcap ((\exists HasInsurance.TheftInsurance) \sqcup (\exists HasInsurance.FireInsurance)))$
$\beta_3 = ((\exists HasAirConditioning.Airconditioning) \sqcap (\exists HasExColor.(ExColorBlack \sqcup ExColorGray)))$
$\beta_4 = (\exists \mathit{price}.\mathit{ls}(\mathit{22000}, \mathit{24000}))$

[8] An investigation on how to compute t_β,t_σ,n_i and m_i is out of the scope of this paper. We can assume they are determined in advance by means of either direct assignment methods (Ordering, Simple Assessing or Ratio Comparison) or pairwise comparison methods (like AHP and Geometric Mean) [18].

$\beta_5 = (\exists km\_warranty.rs(150000, 175000))$
$\mathcal{B} = (0.1 \cdot \beta_1 + 0.2 \cdot \beta_2 + 0.1 \cdot \beta_3 + 0.2 \cdot \beta_4 + 0.4 \cdot \beta_5)[\geqslant 0.7]$

σ is $S = Sedan \sqcap (\geqslant price\ 22000)[\geqslant 1.0]$
$\sigma_1 = ((\exists HasNavigator.NavigatorPack) \rightarrow (\exists Has\ Price.R(22500, 22750))))$
$\sigma_2 = (\exists HasInsurance.InsurancePlus)$
$\sigma_3 = (\exists km\_warranty.ls(100000, 125000))$
$\sigma_4 = (\exists HasMWarranty.L(60, 72))$
$\sigma_5 = ((\exists HasExColor.ExColorBlack) \rightarrow (\exists Has\ AirConditioning.AirConditioning))$
$\mathcal{S} = (0.3 \cdot \sigma_1 + 0.1 \cdot \sigma_2 + 0.3 \cdot \sigma_3 + 0.1 \cdot \sigma_4 + 0.2 \cdot \sigma_5)[\geqslant 0.6]$

Let
$$\bar{\mathcal{K}} = \langle \mathcal{T} \cup \{\sigma, \beta\} \cup \{Buy = (\mathcal{B}[\geqslant t_\beta]), Sell = (\mathcal{S}[\geqslant t_\sigma])\}, \mathcal{R} \rangle$$
Then, it can be verified that the Pareto optimal agreement value is

$$v_P = bsb(\mathcal{K}, Buy \sqcap Sell) = 0.7625\ ,$$

with a Pareto agreement $\bar{\mathcal{I}}$ that maximally satisfies

$$(= HasPrice\ 22500.0) \sqcap (= HasKMWarranty\ 100000.0) \sqcap (= HasMWarranty\ 60.0)\ .$$

i.e., , the car may be sold with a price of 22500, 100000 km warranty and 60 month warranty.

6 Related Work and discussion

Automated bilateral negotiation has been widely investigated, both in artificial intelligence and in microeconomics research communities. AI-oriented research has usually focused on automated negotiation among agents, and on designing high-level protocols for agent interaction [13, 6, 12]. As stated in [15], negotiation mechanisms often involve the presence of a mediator , which collects information from bargainers and exploits them in order to propose an efficient negotiation outcome. Various recent proposals adopt a mediator, including [7, 11, 8]. However in these approaches no semantic relations among issues are investigated. Several recent logic-based approaches to negotiation are based on propositional logic. In [4], Weighted Propositional Formulas (WPF) are used to express agents preferences in the allocation of indivisible goods, but no common knowledge (as our ontology) is present. The work presented here builds on [20], where a basic propositional logic framework endowed of a logical theory was proposed. In [19] the approach was extended and generalized and complexity issues were discussed. We are currently investigating other negotiation protocols, without the presence of a mediator, allowing to reach an agreement in a reasonable amount of communication rounds. The use of aggregate operators is also under investigation.

Acknowledgement

This work has been supported in part by EU-FP6-IST-26896 project and Apulia Region funded projects PE_013 *Innovative models for customer profiling* and PS_092 *DIPIS*. We wish to thank Francesco M. Donini for fruitful suggestions and discussions.

References

1. F. Baader and P. Hanschke. A schema for integrating concrete domains into concept languages. In *proc. of IJCAI-91*, pages 452–457, 1991.
2. K. Binmore. *Fun and Games. A Text on Game Theory*. D.C. Heath and Company, 1992.
3. U. Straccia and F. Bobillo. *fuzzyDL*: An Expressive Fuzzy Description Logic Reasoner. In *Proc. of the 2008 International Conference on Fuzzy Systems (FUZZ-08)*, 2008.
4. S. Bouveret, M. Lemaitre, H. Fargier, and J. Lang. Allocation of indivisible goods: a general model and some complexity results. In *Proc. of AAMAS '05*, pages 1309–1310, 2005.
5. Y. Chevaleyre, U. Endriss, and J. Lang. Expressive power of weighted propositional formulas for cardinal preference modeling. In *Proc. of KR 2006*, pages 145–152, 2006.
6. Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Negotiating over small bundles of resources. In *Proc. of AAMAS '05*, pages 296–302, 2005.
7. S. Fatima, M. Wooldridge, and N.R. Jennings. Optimal agendas for multi-issue negotiation. In *Proc. of AAMAS'03*, pages 129–136, 2003.
8. N. Gatti and F. Amigoni. A decentralized bargaining protocol on dependent continuous multi-issue for approximate pareto optimal outcomes. In *Proc. of AAMAS'05*, pages 1213–1214, 2005.
9. Petr Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.
10. N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, M.J. Wooldridge, and C. Sierra. Automated negotiation: prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 10(2):199 – 215, 2001.
11. M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. In *Proc. of AAMAS'02*, pages 753–757, 2002.
12. S. Kraus. *Strategic Negotiation in Multiagent Environments*. The MIT Press, 2001.
13. A. R. Lomuscio, M. Wooldridge, and N. R. Jennings. A classification scheme for negotiation in electronic commerce. *Int Journal of Group Decision and Negotiation*, 12 (1):31–56, 2003.
14. Thomas Lukasiewicz and Umberto Straccia. Tutorial: Managing uncertainty and vagueness in semantic web languages. In *Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, 2007.
15. J.K. MacKie-Mason and M.P. Wellman. Automated markets and trading agents. In *Handbook of Computational Economics*. North-Holland, 2006.
16. J. F. Nash The Bargaining Problem. *Econometrica*, 18(2):155–162, 1950.
17. Simon Parsons, Carles Sierra, and Nick Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
18. J.C. Pomerol and S. Barba-Romero. *Multicriterion Decision Making in Management*. Kluwer Series in Operation Research. Kluwer Academic, 2000.
19. A. Ragone, T. Di Noia, E. Di Sciascio, and F.M. Donini. A logic-based framework to compute pareto agreements in one-shot bilateral negotiation. In *Proc. of ECAI'06*, 2006.
20. A. Ragone, T. Di Noia, E. Di Sciascio, and F.M. Donini. Propositional- logic approach to one-shot multi issue bilateral negotiation. *ACM SIGecom Exchanges*, 5(5):11–21, 2006.
21. H. Raiffa, J. Richardson, and D. Metcalfe. *Negotiation Analysis - The Science and Art of Collaborative Decision Making*. The Belknap Press of Harvard University Press, 2002.
22. J.S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.
23. H. Salkin and M. Kamlesh. *Foundations of Integer Programming*. North-Holland, 1988.
24. Umberto Straccia. A fuzzy description logic for the semantic web. In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence. Elsevier, 2006.
25. M.P. Wellman. Online marketplaces. In *Practical Handbook of Internet Computing*. CRC Press, 2004.
26. M. Wooldridge and S. Parsons. Languages for negotiation. In *Proc of ECAI '04*, pages 393–400, 2000.
27. Dongmo Zhang and Yan Zhang. A computational model of logic-based negotiation. In *Proc. of the AAAI 06*, pages 728–733, 2006.