Semantic Matchmaking via Non-Monotonic Reasoning: the MaMaS-tng Matchmaking Engine

Tommaso Di Noia, Eugenio Di Sciascio and Francesco M. Donini

SisInfLab – Politecnico di Bari and Università della Tuscia
Email: {t.dinoia; disciascio}@poliba.it; donini@unitus.it

Abstract: In a semantic Web service (SWS) matchmaking process, given a request, it is obvious that not only a list of services should be returned, but also a ranking of compatible SWSs should be provided. Obviously having semantically annotated services, the ranking should be based evaluating semantic similarity between descriptions. Furthermore, the availability of such descriptions makes explanation of rank possible and can provide useful information in order to modify or refine the original request. Here we summarize results obtained on this challenging topic exploiting non-monotonic inferences in Description Logics, with particular reference to their implementation within the MaMaS-tng engine.

Keywords: matchmaking, Semantic web, Web services, ranking, abduction, contraction.

1. The need for a logic-based approach

We start with a description of approaches to resource retrieval, highlighting limitations of non-logical approaches, then discussing the general Knowledge Representation principles that a logical approach may yield, before moving on to the Description Logic (DL) setting we adopt [1]. We refer the reader to [6, 2] for several examples and wider argumentation. First of all, we note that using standard relational database techniques to model a resource retrieval framework, there is a need to completely align the attributes of the offered and requested resources descriptions, in order to evaluate a match. If requests and offers are simple names or strings, the only possible match would be identity, resulting in an all-or-nothing approach to the retrieval process. Vague query answering, proposed by [14], was an initial effort to overcome limitations of relational databases. Classical Information Retrieval can be used, too, thus reverting the search for a matching request to similarity between weighted vectors of stemmed terms, as proposed in [11, 16]. The need to work in someway with approximation and ranking in DL-based approaches to matchmaking has also recently led to adopting fuzzy-DLs, as *e.g.*, in [12, 15] or hybrid approaches, mixing semantics with classical unstructured text information retrieval [10, 13]. A further approach structures resource descriptions as set of words. This formalization allows to evaluate not only identity between sets, but also set-based relations between descriptions, such as inclusion, partial overlap, cardinality of set difference. Modeling resource descriptions as set of words is anyway too much sensible to the choice of words employed to be successfully used: the fixed terminology misses meaning that relate words. Such a problem can be overcome by giving terms a logical and shared meaning through an ontology. Nevertheless set-based approaches already have properties that are fundamental in a matchmaking and retrieval process. If we are searching for a resource described through a set of words, we are also interested in sets including the one we search, as they fulfill the resource to retrieve. Moreover even if there are characteristics of the retrieved resource not elicited in the description of the searched resource, an exact match is still possible because absent information have not to be considered negative. The two statements above may be summarized by the following property:

PROPERTY 1. **Open-world descriptions**. *The absence of a characteristic in the description of a resource to be retrieved should not be interpreted as a constraint of absence. Instead it should be considered as a characteristic that could be either refined later or left open if it is irrelevant for the user searching for the resource.*

Obviously, some specific characteristic might be declared to be closed, as long as such a closure is made piecewise, using some known declarative tool *e.g.*, Autoepistemic DLs [7] or Circumscription in DLs [9]. The set-based match evaluation is non-symmetric: if we search for a resource W_1, whose describing set of words is included in a set characterizing resource W_2, we may consider W_2 a resource perfectly satisfying the request for W_1. On the other hand if we use the description of W_2 for the search, W_1 may also satisfy the request only partially, as some of the terms describing W_2 may be not included in the W_1 set.

PROPERTY 2. **Non-symmetric evaluation**. *Given two semantic-based descriptions W (for semantic Web service) and Q (for Query), a matchmaking system may give different evaluations depending on whether it is trying to match W with Q, or Q with W* — i.e., *depending on who is going to use this evaluation.*

From now on we assume that resource descriptions, requested and offered, are expressed in a DL, equipped with a model-theoretic semantics. This approach includes the sets-of-keywords one, since a set of keywords can be considered also as a conjunction of concept names. We also assume that a common ontology is established, as a TBox $\mathcal{T}$ in DL.

2. Explanation Oriented Matchmaking

DL-based systems usually provide two basic reasoning services for $\mathcal{T}$, namely satisfiability and subsumption.They can be defined, informally, as follows:

Concept Satisfiability: Given an ontology $\mathcal{T}$ (for **T**erminology) modeling the domain we are investigating on, and a descrip-

[1] we assume hereafter the reader be familiar with basics of Description Logics formalisms

tion $\mathtt{Q}$ of a resource referring to the ontology: is the information modeled in the description consistent with the one in the ontology?

Subsumption:Given an ontology $\mathcal{T}$ modeling the domain we are investigating on, and two resources described by expressions –$\mathtt{Q}$, $\mathtt{W}$– referring to the information modeled in the ontology: is the information describing a resource more general than the other one's description?

Both Subsumption and Concept Satisfiability are adequate in all those scenarios where a yes/no answer is enough. For example, given a resource and a request represented respectively by a concept $\mathtt{W}$ and a concept $\mathtt{Q}$, using Concept Satisfiability we are able to determine whether they are compatible,*i.e.*, $\mathtt{W}$ models information which is not in conflict with the one modeled by $\mathtt{Q}$. This task can be performed checking the satisfiability of the concept $\mathtt{W} \sqcap \mathtt{Q}$ with respect to a reference ontology $\mathcal{T}$. On the other hand Subsumption can be used to verify, for example, if a resource described by $\mathtt{W}$ satisfies a request $\mathtt{Q}$. It is easy understandable that if the relation $\mathtt{W} \sqsubseteq \mathtt{Q}$ holds, then $\mathtt{W}$ is more specific than $\mathtt{Q}$ and contains at least all the requested features. In [1, 4] Concept Contraction and Concept Abduction, non-standard inference services for DLs, were introduced and defined. In this subsection we briefly recall their definitions, explaining their rationale and the need for them in resource retrieval.

2.1 Nonmonotonic Inferences

Starting with the concepts $\mathtt{W}$ and $\mathtt{Q}$, if their conjunction $\mathtt{W} \sqcap \mathtt{Q}$ is unsatisfiable in the TBox $\mathcal{T}$ representing the ontology, *i.e.*, they are not compatible with each other, our aim is to retract requirements in $\mathtt{Q}$, G (for *Give up*), to obtain a concept K (for *Keep*) such that $K \sqcap \mathtt{W}$ is satisfiable in $\mathcal{T}$.

DEFINITION 1.*Let $\mathcal{L}$ be a DL, $\mathtt{W}$, $\mathtt{Q}$, be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both $\mathtt{W}$ and $\mathtt{Q}$ are satisfiable in $\mathcal{T}$. A* Concept Contraction Problem *(CCP), identified by $\langle \mathcal{L}, \mathtt{Q}, \mathtt{W}, \mathcal{T} \rangle$, is finding a pair of concepts $\langle G, K \rangle \in \mathcal{L} \times \mathcal{L}$ such that $\mathcal{T} \models \mathtt{Q} \equiv G \sqcap K$, and $\mathcal{T} \models K \sqcap \mathtt{W} \not\equiv \bot$. We call K a* contraction *of $\mathtt{Q}$ according to $\mathtt{W}$ and $\mathcal{T}$.*

We note that there is always the trivial solution $\langle G, K \rangle = \langle \mathtt{Q}, \top \rangle$ to a CCP[2]. This solution corresponds to the most drastic contraction, that gives up everything of $\mathtt{Q}$. In our resource retrieval framework, it models the (infrequent) situation in which, in front of some very appealing resource $\mathtt{W}$, incompatible with the requested one, a user just gives up completely his/her specifications $\mathtt{Q}$ in order to meet $\mathtt{W}$. At a first glance it would seem more reasonable the solution $\langle G, K \rangle = \langle \mathtt{Q}, \bot \rangle$ instead of $\langle G, K \rangle = \langle \mathtt{Q}, \top \rangle$. Actually, we can read the former solution as "*After giving up everything in my original query, I want nothing*" and the latter as "*After giving up everything in my original query, whatever is good for me*". In a matchmaking scenario, this solution is the most correct one. On the other hand, when $\mathtt{W} \sqcap \mathtt{Q}$ is satisfiable in $\mathcal{T}$, the "best" possible solution is $\langle \top, \mathtt{Q} \rangle$, that is, give up nothing – if possible. Since usually one wants to give up as few things as possible, some minimality in the contraction must be defined [8, 1, 3].

[2] With $\top$ we refer to the most generic concept in an ontology. We use $\bot$ to denote the most specific concept (the unsatisfiable concept). In OWL words, they are represented by `<owl:Thing>` and `<owl:Nothing>` respectively.

If the SWS description $\mathtt{W}$ and the query $\mathtt{Q}$ are compatible with each other, the partial specifications problem still holds, that is, it could be the case that $\mathtt{W}$ – though compatible – does not imply $\mathtt{Q}$. Using DL syntax we write: $\mathcal{T} \models \mathtt{Q} \sqcap \mathtt{W} \not\sqsubseteq \bot$ and $\mathcal{T} \models \mathtt{W} \not\sqsubseteq \mathtt{Q}$. Then, it is necessary to assess what should be hypothesized (H) in $\mathtt{W}$ in order to completely satisfy $\mathtt{Q}$.

DEFINITION 2.*Let $\mathcal{L}$ be a DL, $\mathtt{W}$, $\mathtt{Q}$, be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both $\mathtt{W}$ and $\mathtt{Q}$ are satisfiable in $\mathcal{T}$. A* Concept Abduction Problem *(CAP), identified by $\langle \mathcal{L}, \mathtt{Q}, \mathtt{W}, \mathcal{T} \rangle$, is finding a concept $H \in \mathcal{L}$ such that $\mathcal{T} \models \mathtt{W} \sqcap H \sqsubseteq \mathtt{Q}$, and moreover $\mathtt{W} \sqcap H$ is satisfiable in $\mathcal{T}$. We call H a* hypothesis *about $\mathtt{W}$ according to $\mathtt{Q}$ and $\mathcal{T}$.*

Observe that in the definition, we limit to satisfiable $\mathtt{W}$ and $\mathtt{Q}$, since $\mathtt{Q}$ unsatisfiable implies that the CAP has no solution at all, while $\mathtt{W}$ unsatisfiable leads to counterintuitive results ($\neg\mathtt{Q}$ would be a solution in that case). If $\mathtt{W} \sqsubseteq \mathtt{Q}$ then we have $H = \top$ as a solution to the related CAP. Hence, Concept Abduction extends subsumption. On the other hand, if $\mathtt{W} \equiv \top$ then $\mathcal{T} \models H \sqsubseteq \mathtt{Q}$. Notice that both Concept Abduction and Concept Contraction can be used for respectively subsumption and satisfiability explanation. For Concept Contraction, having two concepts not compatible with each other, in the solution $\langle G, K \rangle$ to the CCP $\langle \mathcal{L}, \mathtt{Q}, \mathtt{W}, \mathcal{T} \rangle$, G represents "why" $\mathtt{Q} \sqcap \mathtt{W}$ are not compatible*i.e.*, which part of the query is conflicting with SWS description. For Concept Abduction, having $\mathtt{Q}$ and $\mathtt{W}$ such that $\mathcal{T} \models \mathtt{W} \not\sqsubseteq \mathtt{Q}$, the solution H to the CAP $\langle \mathcal{L}, \mathtt{Q}, \mathtt{W}, \mathcal{T} \rangle$ represents "why" the subsumption relation does not hold. H can be interpreted as *what is requested in $\mathtt{Q}$ and not specified in $\mathtt{W}$.*

2.2 Logic-Based Matchmaking via Concept Abduction and Concept Contraction

Both Concept Abduction and Concept Contraction can be used to suggest guidelines on what, given an offered resource $\mathtt{W}$, has to be revised and/or hypothesized to obtain a full match with the request. Let us suppose to have a request $\mathtt{Q}$, a resource $\mathtt{W}$ and an ontology $\mathcal{T}$ such that $\mathcal{T} \models \mathtt{Q} \sqcap \mathtt{W} \not\sqsubseteq \bot$, *i.e.*, they are incompatible with each other. In order to gain compatibility, a Concept Contraction is needed so that giving up G in $\mathtt{Q}$, the remaining K could be satisfied by $\mathtt{W}$. Now, if $\mathcal{T} \not\models \mathtt{W} \sqsubseteq K$, the solution H_K to the CAP $\langle \mathcal{L}, K, \mathtt{W}, \mathcal{T} \rangle$ represents what is in K and is not specified in $\mathtt{W}$. As the $\mathtt{W}$ obtained is an approximated match of $\mathtt{Q}$, then evaluating how good is the approximation would be extremely useful. Given more than one resource, which is the best approximation? How a numerical score can be assigned to the approximation, based on K,H and G, in order to rank the resources? Algorithm 8 provides answers to the raised issues.

[lines 1-4] Having a request $\mathtt{Q}$ and an offered service $\mathtt{W}$, if their descriptions conjunction is not satisfiable w.r.t. the ontology they refer to (i.e., they are not compatible with each other for some concepts in their descriptions), first a contraction on $\mathtt{Q}$ is performed in order to regain compatibility [line 2] and then [line 3] we compute what has to be hypothesized in $\mathtt{W}$ in order to completely satisfy $\mathtt{Q}$ (its contraction). The returned values represent:

$\langle G, K \rangle$: The first item is what has to be given up in the request – G – in order to continue the process, or, in other words, why

Algorithm: $explain(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{L})$

Input: $\mathtt{W}, \mathtt{Q}$ concepts in $\mathcal{L}$ such that $\mathcal{T} \not\models \mathtt{W}$ and $\mathcal{T} \not\models \mathtt{Q}$

Output: $\langle G, K\rangle$, H *i.e.*, the part in $\mathtt{Q}$ that should be retracted G and kept K and the part in $\mathtt{W}$ that should be hypothesized to find a full match between $\mathtt{W}$ and $\mathtt{Q}$

1 **if** $\mathcal{T} \models \mathtt{Q} \sqcap \mathtt{W} \sqsubseteq \perp$ **then**

2 $\quad\langle G, K\rangle = contract(\mathtt{W}, \mathtt{Q}, \mathcal{T})$;

3 $\quad H_K = abduce(\mathtt{W}, K, \mathcal{T})$;

4 $\quad$**return** $\langle G, K\rangle, H_K$;

5 **else**

6 $\quad H = abduce(\mathtt{W}, \mathtt{Q}, \mathcal{T})$;

7 $\quad$**return** $\langle \top, \mathtt{Q}\rangle, H$;

8 **end**

Algorithm 1:

$\mathtt{Q}$ is not compatible with $\mathtt{W}$. The second item is the contracted request K that is no more in conflict with the request.
H_K: After the contraction of $\mathtt{Q}$, the request is represented by K, i.e. the portion of $\mathtt{Q}$ that is compatible with $\mathtt{W}$. H_K represents what has to be hypothesized in $\mathtt{W}$ in order to completely satisfy K, or, in other words, why $\mathtt{W}$ does not completely satisfy K.
[lines 5-7] If the conjunction of $\mathtt{Q}$'s and $\mathtt{W}$'s description is satisfiable w.r.t. the ontology they refer to, then no contraction is needed and only an abductive process is carried out.
The algorithm $explain$ returns values useful in a retrieval system where explanation of the results is needed and/or a belief revision process is admitted.

EXAMPLE 1.*As a simple example, suppose you are looking for "smoking room and price includes a Wi-Fi connection" and you find two web services offering respectively "bedrooms and price includes a cable Internet connection" and "smoking twin rooms with Internet connection included". It is easy to see that both semantic web service descriptions do not match completely the request. The question is: why? How to automatically compute the reasons why the query is not matched by the web service descriptions? In formulas, w.r.t. to the ontology in Figure 1, we can represent the above query and SWS descriptions as:*

$\mathtt{Q} = \mathtt{SmokingRoom} \sqcap \forall \mathtt{incl.WiFi}$

$\mathtt{W}_1 = \mathtt{Bedroom} \sqcap \exists \mathtt{incl} \sqcap \forall \mathtt{incl.CableConnection}$

$\mathtt{W}_2 = \mathtt{TwinRoom} \sqcap \mathtt{SmokingRoom} \sqcap \exists \mathtt{incl} \sqcap \forall \mathtt{incl.InternetConnection}$

If we consider $\mathtt{Q}$ and $\mathtt{W}_1$ we see that they are inconsistent with each other: $\mathcal{T} \models \mathtt{Q} \sqcap \mathtt{W}_1 \sqsubseteq \perp$. If we use $explain(\mathtt{W}_1, \mathtt{Q}, \mathcal{T}, \mathcal{L})$, the algorithm recognizes the inconsistency (row 1) and computes a contracted version of $\mathtt{Q}$ in line 2. In this case a possible contraction would be:

$$\langle G, K\rangle = \langle \forall \mathtt{incl.WiFi}, \mathtt{SmokingRoom} \sqcap \forall \mathtt{incl.InternetConnection}\rangle$$

In other words the algorithm is suggesting: "if you are interested in $\mathtt{W}_1$ you have to know that you surely will not have a Wi-Fi connection. I suggest to contract your request giving up your Wi-Fi specification".

With respect to the contracted query (represented by K) $explain(\mathtt{W}_1, \mathtt{Q}, \mathcal{T}, \mathcal{L})$ tries to find what is underspecified in $\mathtt{W}_1$ solving a concept abduction problem [line 3].

$$H = \forall \mathtt{guest.Smoker}$$

This result can be read as "based on what is explicitly stated in $\mathtt{W}_1$ I cannot establish if the rooms are smoking rooms or no smoking ones. I have no information on this". Summing up, $explain(\mathtt{W}_1, \mathtt{Q}, \mathcal{T}, \mathcal{L})$ explains the reasons why $\mathtt{W}_1$ is not a full match for $\mathtt{Q}$ saying:
- "There is something in $\mathtt{W}_1$ which is explicitly in conflict with your $\mathtt{Q}$":

$$G = \forall \mathtt{incl.WiFi}$$

- "There is something you requested that is not specified in any way in $\mathtt{W}_1$. There are some missing information":

$$H = \forall \mathtt{guest.Smoker}$$

In a similar way, if we consider $\mathtt{Q}$ and $\mathtt{W}_2$ we see that they are not conflicting with each other. Hence, there is nothing to give up and then $explain(\mathtt{W}_2, \mathtt{Q}, \mathcal{T}, \mathcal{L})$ in line 6 computes:

$$H = \forall \mathtt{incl.WiFi}$$

That is, based on the description of $\mathtt{W}_2$, it is not possible to establish if the Wi-Fi connection is included or not.

The algorithm $explain$ does not depend on the particular DL adopted. Based on the minimality criteria proposed in [1] the length H of the solution to a CAP for an $\mathcal{ALN}$ DL can be computed as proposed in [5]. Hence, a relevance ranking score can be computed by an utility function defined as $U(G, K, H_K)$.

$$
\begin{aligned}
\mathtt{InternetConnection} &\sqsupseteq \mathtt{CableConnection} \sqcup \mathtt{WiFi} \\
& \quad disj(\mathtt{WiFi}, \mathtt{CableConnection}) \\
\mathtt{SwimmingPool} \sqcup \mathtt{SPA} &\sqsubseteq \mathtt{FitnessFacilities} \\
\mathtt{SAT\text{-}TV} &\sqsubseteq \mathtt{TV} \\
\mathtt{HotelFacilities} &\sqsupseteq \mathtt{TV} \sqcup \mathtt{FitnessFacilities} \sqcup \\
& \quad \sqcup \mathtt{Breakfast} \sqcup \mathtt{HotelFacilities} \\
\mathtt{Bedroom} &\equiv \exists \mathtt{bed} \sqcap \exists \mathtt{guest} \\
\mathtt{SingleRoom} &\equiv \mathtt{Bedroom} \sqcap (\leq 1 \mathtt{bed}) \sqcap \\
& \quad \sqcap (\leq 1 \mathtt{guest}) \\
\mathtt{DoubleRoom} &\equiv \mathtt{Bedroom} \sqcap (\leq 1 \mathtt{bed}) \sqcap \\
& \quad \sqcap (\leq 2 \mathtt{guest}) \\
\mathtt{TwinRoom} &\equiv \mathtt{Bedroom} \sqcap (= 2 \mathtt{bed}) \sqcap \\
& \quad \sqcap (\leq 2 \mathtt{guest}) \\
\mathtt{SmokingRoom} &\equiv \mathtt{Bedroom} \sqcap \forall \mathtt{guest.Smoker} \\
\mathtt{NoSmokingRoom} &\equiv \mathtt{Bedroom} \sqcap \forall \mathtt{guest.}\neg\mathtt{Smoker}
\end{aligned}
$$

Fig. 1. The reference ontology. $disj(\mathtt{WiFi}, \mathtt{CableConnection})$ represents $\mathtt{WiFi} \sqsubseteq \neg\mathtt{CableConnection}$ and $\mathtt{CableConnection} \sqsubseteq \neg\mathtt{WiFi}$.

3. Dealing with User Preferences

In a semantic discovery process, a user query, can be split often into two separate parts: **strict** requirements and **preferences**. **Strict** requirements represent what, in the query, has to be strictly matched by the semantic web service description. **Preferences** can be seen as soft user requirements. In other words, the user will accept even a web service whose functionalities do not provide exactly the ones represented by a preference. Usually, a weight is associated to each preference in order to represent its worth (absolute or relative to the other preferences). Hence, for a user query $\mathtt{Q}$ we distinguish between a concept $\mathtt{Q}_S$ representing strict requirements and a set of weighted concepts $\langle \mathtt{Q}, v\rangle$ where $\mathtt{Q}$ is a DL concept and v is a numerical value representing preference worth. It should be clear that a matchmaking process has not to be performed w.r.t. $\mathtt{Q}_S$. It represents what the user is not willing to risk on at all. He does not want to hypothesize nothing on it. An approximate solution would not be significant for $\mathtt{Q}_S$. On the other hand, even though a preference is satisfied *with a certain degree* (not necessarily completely) the user will be satisfied *with a certain degree* as well. Given an ontology $\mathcal{T}$, a semantic web service description $\mathtt{W}$, a strict requirements query $\mathtt{Q}_S$ and a set of preferences $\mathcal{P} = \{\langle \mathtt{Q}_i, v_i\rangle\}$ we compute a global ranking score using Algorithm 2. Here we retrieve, *i.e.,* assign $score \geq 0$, only those web services whose description fully satisfies user strict requirements. Once we have a web service description such that $\mathcal{T} \models \mathtt{W} \sqsubseteq \mathtt{Q}_S$, then we compute how much it satisfies user preferences. For each preference we take into account both $U(explain(\mathtt{W}, \mathtt{Q}_i, \mathcal{T}, \mathcal{L}))$ *i.e.*, the similarity degree computed using non-standard reasoning, and the value expressed by the user to represent preference worth.

```
Algorithm: preference_retrieve(W, Q_S, P, T, L)
score = 0 ;
if T |= W ⊑ Q_S then
    foreach ⟨Q_i, v_i⟩ ∈ P do
        score = score + v_i · U(explain(W, Q_i, T, L));
    end
end
return score;
```

Algorithm 2:

4. Illustrative example

In order to describe how $preference\_retrieve(\mathtt{W}, \mathtt{Q}_S, \mathcal{P}, \mathcal{T}, \mathcal{L})$ works, in Figure 2 we model a simple hotel booking scenario. Strict requirements $\mathtt{Q}_S$, preferences and semantic web service descriptions $\mathcal{P} = \{\langle \mathtt{Q}_1, v_1\rangle, \langle \mathtt{Q}_2, v_2\rangle, \langle \mathtt{Q}_3, v_3\rangle\}$, $\mathtt{W}_1, \mathtt{W}_2$ and $\mathtt{W}_3$ are modeled with respect to the toy ontology $\mathcal{T}$ reported in Figure 1. It is easy to see that $\mathcal{T} \models \mathtt{W}_1 \sqsubseteq \mathtt{Q}_S$; $\mathcal{T} \models \mathtt{W}_2 \sqsubseteq \mathtt{Q}_S$; $\mathcal{T} \not\models \mathtt{W}_3 \sqsubseteq \mathtt{Q}_S$. Since $\mathtt{W}_3$ does not satisfy strict requirements $\mathtt{Q}_S$, differently from $\mathtt{W}_1$ and $\mathtt{W}_2$, it will not be selected for the matchmaking process. Then we proceed applying algorithm $retrieve$ to remaining available services. We recall that $retrieve$ returns a 3-tuple whose first item represents what is incompatible (and should be given up) in the request w.r.t. the service, while the second item represents what is compatible and can be kept in the request. The third item corresponds to what has to be hypothesized (to get a full match) w.r.t. the already contracted part of the request. With respect to Figure 2 we can say: **(1)** $\mathtt{W}_1$ is compatible with $\mathtt{Q}_1$. No contraction of preference specification is needed. In fact, $\langle G_{1,1}, K_{1,1}\rangle = \langle \top, \mathtt{Q}_1\rangle$. Since $\mathtt{W}_1$ completely satisfies $\mathtt{Q}_1$ no specific hypotheses have to be formulated. Then $H = \top$; **(2)** $\mathtt{W}_1$ is not compatible with $\mathtt{Q}_2$. Specification on smoking cannot be satisfied. With respect to the contracted preference, in $\mathtt{W}_1$ nothing is specified about SAT TV; **(3)** $\mathtt{W}_1$ is compatible with $\mathtt{Q}_3$. No contraction of preference specification is needed. In $\mathtt{W}_1$ nothing is specified about Wi-Fi connection; **(4)** $\mathtt{W}_2$ is not compatible with $\mathtt{Q}_1$. With respect to the definition of `TwinRoom` given in the ontology, the specification related to the minimum number of beds has to be given up. In fact, since $\mathtt{W}_2$ is offering double rooms, then you could also have 2 beds in your room. Finally, $\mathtt{W}_2$ completely satisfies the contracted preference specification $K_{2,1} = (\leq 2\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap (\leq 2\mathtt{guest})$; **(5)** $\mathtt{W}_2$ is compatible with $\mathtt{Q}_2$. Unfortunately, in $\mathtt{W}_2$ is not specified if smoking is allowed or not; **(6)** $\mathtt{W}_2$ is compatible with $\mathtt{Q}_3$. Since $\mathtt{W}_1$ completely satisfies $\mathtt{Q}_1$ no specific hypotheses have to be formulated.

5. How to compute $U(explain(\mathbf{W}, \mathbf{Q}, \mathcal{T}, \mathcal{L}))$: the `MaMaS` way

`MaMaS` (**Ma**tch**Ma**king **S**ervice) is a DL reasoner for $\mathcal{ALN}$ implementing algorithms for non-monotonic reasoning and semantic-based ranking. It is able to solve both concept abduction problems and concept contraction ones in $\mathcal{ALN}$. Using `MaMaS`, given an ontology $\mathcal{T}$, it is also possible to rank a WS description w.r.t. a query using $rankPotential$ algorithm as described in [5]. In a nutshell, given two concept descriptions $\mathtt{Q}$ and $\mathtt{W}$, $rankPotential$ compares their normalized forms and evaluate a score representing the length of the corresponding concept abduction problem.

rankPotential. Given two $\mathcal{ALN}$ concepts $\mathtt{W}$ and $\mathtt{Q}$ in normal form w.r.t. an ontology $\mathcal{T}$ in $\mathcal{ALN}$, $rankPotential(\mathtt{W}, \mathtt{Q})$ computes the "length" of a solution to a concept abduction problem [6]. Note that given an ontology $\mathcal{T}$ and a query $\mathtt{Q}$, the maximum value for $rankPotential(\mathtt{W}, \mathtt{Q})$ is computed when $\mathcal{T} \models \mathtt{W} \equiv \top$. Hence, given a query $\mathtt{Q}$ and a concept description C, the value resulting from $rankPotential(\top, \mathtt{Q})$ can be used to normalize $rankPotential(C, \mathtt{Q})$. In the following we show how to use $rankPotential$ in order to compute a value representing a similarity degree of a WS description $\mathtt{W}$ and a query $\mathtt{Q}$ both normalized w.r.t. an ontology $\mathcal{T}$. We compute N, g and h as: $N = rankPotential(\top, \mathtt{Q})$, represents the maximum score computed by $rankPotential$ associated to the solution of a concept abduction problem, given a query $\mathtt{Q}$ and an ontology $\mathcal{T}$; $g = rankPotential(\mathtt{Q}, K)$ (where K is the result of $retrieve(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{ALN})$), represents a how much of $\mathtt{Q}$ has been given up in K. In other words g assigns a score to G resulting from $retrieve(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{ALN})$; $h = rankPotential(\mathtt{W}, K)$, represents how much of K, resulting from $retrieve(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{ALN})$, is not specified in $\mathtt{W}$. It assigns a score to H solution of $retrieve(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{ALN})$. Based on N, g and h a possible formulation of $U(retrieve(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{ALN}))$ is: $U = (1 - \frac{g}{N}) \cdot (1 - \frac{h}{N})$. Notice that even though H, does not explicitly appear in the computation of U it is implicitly used in the computation of h.

6. Illustrative example (cont'd)

In order to show how to use $rankPotential$ to compute $U(retrieve(\mathtt{W}, \mathtt{Q}, \mathcal{T}, \mathcal{ALN}))$, let us consider $\mathtt{W}_1$, $\mathtt{W}_2$ and D_1 and $\mathtt{Q}_2$ in Section 4 and compute their normalized versions as shown in Figure 2. Since $\mathcal{T} \not\models \mathtt{Q}_1 \sqcap \mathtt{W}_1 \sqsubseteq \bot$, then we do not have to contract $\mathtt{Q}_1$ w.r.t. $\mathtt{W}_1$ (see Section 4) and then we have $g_{1,1} = 0$. On the other side, since $\mathcal{T} \models \mathtt{W}_1 \sqsubseteq \mathtt{Q}_1$ then $h_{1,1} = 0$. Differently form the previous case, $\mathcal{T} \models \mathtt{Q}_1 \sqcap \mathtt{W}_1 \sqsubseteq \bot$. Hence, we contract $\mathtt{Q}_1$ and we have as a result $\langle G_{2,1}, K_{2,1}\rangle = contract(\mathtt{W}_2, \mathtt{Q}_1, \mathcal{T})$.

$K^{norm}_{2,1} = (\leq 1\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap (\leq 2\mathtt{guest})$; $g_{2,1} = 1$; $h_{2,1} = 0$

Since $\mathcal{T} \models \mathtt{Q}_2 \sqcap \mathtt{W}_1 \sqsubseteq \bot$, then we contract $\mathtt{Q}_2$ w.r.t. $\mathtt{W}_1$ and we have $\langle G_{1,2}, K_{1,2}\rangle = contract(\mathtt{W}_1, \mathtt{Q}_2, \mathcal{T})$ (see Section 4).

$$
\begin{aligned}
\mathtt{Q}_S &= \mathtt{Bedroom} \sqcap \forall \mathtt{incl}.\forall \mathtt{facility}.\mathtt{InternetConnection} \quad \textit{“I want to book a bedroom whose price includes the use of the Internet”.}\\
\mathtt{Q}_1 &= \mathtt{TwinRoom} \quad \textit{“It would be nice if it was a twin room”.}\\
\mathtt{Q}_2 &= \mathtt{NoSmokingRoom} \sqcap \forall \mathtt{incl}.\forall \mathtt{facility}.\mathtt{SAT\text{-}TV} \quad \textit{“I would prefer a no smoking room with SAT TV. I would prefer a no smoking room”.}\\
\mathtt{Q}_3 &= \forall \mathtt{incl}.\forall \mathtt{facility}.\mathtt{WiFi} \quad \textit{“A Wi-Fi connection would be appreciated”.}\\
\mathtt{W}_1 &= \mathtt{TwinRoom} \sqcap \exists \mathtt{incl} \sqcap \forall \mathtt{incl}.(\exists \mathtt{facility} \sqcap \forall \mathtt{facility}.(\mathtt{TV} \sqcap \mathtt{CableConnection})) \sqcap\\
&\quad \mathtt{SmokingRoom} \quad \textit{“Booking twin rooms. Price includes Internet connection and cable TV. Smoking is allowed”.}\\
\mathtt{W}_2 &= \mathtt{DoubleRoom} \sqcap \exists \mathtt{incl} \sqcap \forall \mathtt{incl}.(\exists \mathtt{facility} \sqcap \forall \mathtt{facility}.(\mathtt{SAT\text{-}TV} \sqcap \mathtt{WiFi}))\\
&\quad \textit{“Booking double rooms. Price includes Wi-Fi Internet connection and SAT TV.”}\\
\mathtt{W}_3 &= \mathtt{SingleRoom} \sqcap \exists \mathtt{facility} \sqcap \forall \mathtt{facility}.(\mathtt{SPA} \sqcap \mathtt{WiFi}) \quad \textit{“Single rooms. The hotel has a SPA and Wi-Fi Internet connection”.}
\end{aligned}
$$

$$
\begin{aligned}
explain(\mathtt{W}_1, \mathtt{Q}_1, \mathcal{T}, \mathcal{L}) &= \langle \top, \mathtt{Q}_1 \rangle, \top && (1)\\
explain(\mathtt{W}_1, \mathtt{Q}_2, \mathcal{T}, \mathcal{L}) &= \langle \forall \mathtt{guest}.\neg \mathtt{Smoker}, (\geq 1\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap \forall \mathtt{incl}.\forall \mathtt{facility}.\mathtt{SAT\text{-}TV} \rangle, \forall \mathtt{incl}.\forall \mathtt{facility}.\mathtt{SAT\text{-}TV} && (2)\\
explain(\mathtt{W}_1, \mathtt{Q}_3, \mathcal{T}, \mathcal{L}) &= \langle \top, \mathtt{Q}_3 \rangle, \forall \mathtt{incl}.\forall \mathtt{facility}.\mathtt{WiFi} && (3)\\
explain(\mathtt{W}_2, \mathtt{Q}_1, \mathcal{T}, \mathcal{L}) &= \langle (\geq 2\mathtt{bed}), (\leq 2\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap (\leq 2\mathtt{guest}) \rangle, \top && (4)\\
explain(\mathtt{W}_2, \mathtt{Q}_2, \mathcal{T}, \mathcal{L}) &= \langle \top, \mathtt{Q}_2 \rangle, \forall \mathtt{guest}.\neg \mathtt{Smoker} && (5)\\
explain(\mathtt{W}_2, \mathtt{Q}_3, \mathcal{T}, \mathcal{L}) &= \langle \top, \mathtt{Q}_3 \rangle, \top && (6)
\end{aligned}
$$

$$
\begin{aligned}
\mathtt{Q}_1^{norm} &= (\geq 2\mathtt{bed}) \sqcap (\leq 2\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap (\leq 2\mathtt{guest}); \;\; N_1 = 4\\
\mathtt{Q}_2^{norm} &= (\geq 1\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap \forall \mathtt{guest}.\neg \mathtt{Smoker} \sqcap \forall \mathtt{incl}.\forall \mathtt{facility}.(\mathtt{SAT\text{-}TV} \sqcap \mathtt{TV} \sqcap \mathtt{HotelFacilities}); \;\; N_2 = 6\\
\mathtt{W}_1^{norm} &= (\geq 2\mathtt{bed}) \sqcap (\leq 2\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap (\leq 2\mathtt{guest}) \sqcap \forall \mathtt{guest}.\mathtt{Smoker} \sqcap (\geq 1\mathtt{incl}) \sqcap \forall \mathtt{incl}.((\geq 1\mathtt{facility}) \sqcap\\
&\quad \sqcap \forall \mathtt{facility}.(\mathtt{TV} \sqcap \mathtt{HotelFacilities} \sqcap \mathtt{CableConnection} \sqcap \neg \mathtt{WiFi} \sqcap \mathtt{InternetConnection}))\\
\mathtt{W}_2^{norm} &= (\geq 1\mathtt{bed}) \sqcap (\leq 1\mathtt{bed}) \sqcap (\leq 2\mathtt{guest}) \sqcap (\geq 1\mathtt{guest}) \sqcap (\geq 1\mathtt{incl}) \sqcap \forall \mathtt{incl}.((\geq 1\mathtt{facility}) \sqcap\\
&\quad \sqcap \forall \mathtt{facility}.(\mathtt{SAT\text{-}TV} \sqcap \mathtt{TV} \sqcap \mathtt{HotelFacilities} \sqcap \mathtt{WiFi} \sqcap \neg \mathtt{CableConnection} \sqcap \mathtt{InternetConnection}))
\end{aligned}
$$

Fig. 2. User request and service descriptions in the reference scenario; Explanation results in the matchmaking process; Normalized version of $\mathtt{Q}_1$, $\mathtt{Q}_2$ (with corresponding values N_1 and N_2) $\mathtt{W}_1$ and $\mathtt{W}_2$.

$K_{1,2}^{norm} = (\geq 1\mathtt{bed}) \sqcap (\geq 1\mathtt{guest}) \sqcap \forall \mathtt{incl}.\forall \mathtt{facility}.(\mathtt{SAT\text{-}TV} \sqcap \sqcap \mathtt{TV} \sqcap \mathtt{HotelFacilities})$; $g_{1,2} = 1$; $h_{1,2} = 2$.
Since $\mathcal{T} \not\models \mathtt{Q}_2 \sqcap \mathtt{W}_2 \sqsubseteq \bot$, then we do not have to contract $\mathtt{Q}_2$ w.r.t. $\mathtt{W}_2$ (see Section 4) and then we have $g_{2,2} = 0$ and $h_{2,2} = 1$.

7. Conclusions

This paper has presented main characteristics and peculiarities of MaMaS-tng, a semantic matchmaking engine in the $\mathcal{ALN}$ subset of OWL-DL, currently fully operational, endowed of standard and nonmonotonic services, which allows computing both logical rankings and explanations on match rankings. Current work is ongoing on the design and implementation of a novel matchmaking engine, tableu-based, in $\mathcal{ALC}$.

References

[1] S Colucci, T Di Noia, E Di Sciascio, F Donini and M Mongiello, Concept Abduction and Contraction in Description Logics. Proc. of DL 2003, Vol. 81 of CEUR Workshop Proceedings, 2003.

[2] S Colucci, T Di Noia, A Pinto, A Ragone, M Ruta and E Tinelli, A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces. International Journal of Electronic Commerce, Vol. 12, No. 2, 2007, pp. 127–154.

[3] T Di Noia, E Di Sciascio and F Donini, Extending Semantic-Based Matchmaking via Concept Abduction and Contraction, Proc. of EKAW, Lecture Notes in Artificial Intelligence. 2004.

[4] T Di Noia, E Di Sciascio, F Donini and M Mongiello, Abductive matchmaking using description logics. Proc. of IJCAI'03 2003, pp. 337–342.

[5] T Di Noia, E Di Sciascio, F Donini and M Mongiello, A system for principled Matchmaking in an electronic marketplace. International Journal of Electronic Commerce, Vol. 8, No. 4, 2004.

[6] T Di Noia, E D Eugenio and F Donini, Semantic matchmaking as non-monotonic reasoning: A description logic approach. Journal of Artificial Intelligence Research, Vol. 29, 2007, pp. 269–307.

[7] F M Donini, D Nardi and R Rosati, Autoepistemic description logics. Proc. of IJCAI '97 1997, pp. 136–141.

[8] P Gärdenfors, Knowledge in Flux: Modeling the Dynamics of Epistemic States, Bradford Books, MIT Press, 1988.

[9] S Grimm and P Hitzler, Semantic matchmaking of web resources with local closed-world reasoning. International Journal of Electronic Commerce, Vol. 12, No. 2, 2007, pp. 89–126.

[10] M Klusch, B Fries and K Sycara, Automated semantic web service discovery with owls-mx. In AAMAS'06 2006, pp. 915–922.

[11] D Kuokka and L Harada, Integrating Information Via Matchmaking. J. of Intelligent Information Systems, Vol. 6, 1996, pp. 261–279.

[12] U Kuster, B Konig-Ries, M Klein and M Stern, DIANE A Matchmaking-Centered Framework for Automated Service Discovery, Composition, Binding and Invocation on the Web. International Journal of Electronic Commerce, Vol. 12, No. 2, 2007, pp. 41–68.

[13] R Lara, M Corella and P Castells, A flexible model for the location of services on the Web. International Journal of Electronic Commerce, Vol. 12, No. 2, 2007, pp. 11–40.

[14] A Motro, VAGUE: A User Interface to Relational Databases that Permits Vague Queries. ACM TOIS, Vol. 6, No. 3, 1988, pp. 187–214.

[15] A Ragone, U Straccia, T Di Noia, E Di Sciascio and F Donini, Vague knowledge-bases for matchmaking in p2p e-marketplaces, Proc. of ESWC 2007, Vol. 4519 of LNCS, 2007, pp. 414–428.

[16] K Sycara, S Widoff, M Klusch and J Lu, LARKS: Dynamic Matchmaking Among Heterogeneus Software Agents in Cyberspace. Autonomous agents and multi-agent systems, Vol. 5, 2002, pp. 173–203.