Knowledge Elicitation for Query Refinement in a Semantic-Enabled E-Marketplace

Simona Colucci, Tommaso Di Noia,
Eugenio Di Sciascio, Azzurra Ragone
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Via Re David, 200
70125 BARI, Italy
s.colucci,t.dinoia,disciascio,a.ragone@poliba.it

Francesco M. Donini
Universitá della Tuscia
Via San Carlo 32
01100 Viterbo, Italy
donini@unitus.it

ABSTRACT

In this paper we present a knowledge-based approach to the elicitation of information from advertisements, in the framework of a semantic-enabled marketplace. The elicited information can be used for advertisements enriching and refining, without requiring users thorough knowledge of the domain, and to determine a logic-based exact match. The approach exploits non-standard inference services in Description Logics, namely Abduction and Contraction, to tackle a typical problem of semantic-enabled marketplaces, that is the difficulty the average or casual user has in exploiting all the knowledge expressed in an e-commerce domain, which appears necessary to issue requests.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Decision Support; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages

General Terms

Algorithms, Languages, Economics

Keywords

E-commerce, Matchmaking, Description Logics, Concept Abduction, Concept Contraction, knowledge management, Semantic Web

1. INTRODUCTION

Eugenio is planning a well-deserved vacation with his girlfriend and asks his semantic-based personal planner agent to search and book, on his behalf, a reservation according to the following specifications:

"I'm looking for a bedroom for two persons in an hotel at least four stars. Use of SPA included".

The agent discovers the following currently available offers:

- *Double room, smoking permitted. Hotel have an excellent swimming pool available for the exclusive use of their guests.*
- *Wedding room. Smoking is not permitted, price includes breakfast and use of SPA.*
- *Wedding room in a three stars hotel . The price includes breakfast.*

Based on the semantic description it got from Eugenio's request, it chooses the second one, which best fulfills it. When Eugenio gets to the place he discovers the resort is a smoke-free facility. Alas...he is a smoker. Yes it is a bad habit, yet this is not the day to quit; he gets more and more nervous ank keeps smoking outside the resort. While smoking he keeps thinking..."what went wrong?" The answer is that, though his request perfectly fulfilled the available description of the offered resort, the offer had some further specifications he was not aware of, and his personal assistant agent obviously treated –open world assumption– the smoke-free specification as a "don't care" one. Obviously next time Eugenio will be more careful and add to his strict requirements "smoking permitted". But it will be next time...

What got lost in the matchmaking process between the request and the offers? The agent should have informed Eugenio that available offers included what he had specified in his request description, but also other features that are typical of the knowledge domain under consideration – hotels and resorts– that we call for short *bonuses*, *i.e.*, further characteristics not required in the demand and available in the supplies. Yet Eugenio is a computer science professor not an hotel-booking expert, and he had concentrated his request on the basis of his own partial knowledge of the domain. The alternative would have been presenting him the whole domain model, either in a database style form or in a structured ontology. In both cases the process would get more and more boring. But wait, if our system was able to present also features that are unspecified in the request but elicitable from the offers description, it may help Eugenio in rewriting his request taking into account also previously unknown requirements, which he has discovered, and he now considers important to specify.

In this paper we present an approach, and devise algorithms, to deal with scenarios such as the one exemplified above, in the framework of a semantic-based electronic marketplace. In this way we manage to tackle some emerging problems in such e-marketplaces, *i.e.*, the difficulty the average or casual user has in exploiting all the knowledge expressed in an e-commerce domain, which appears necessary to issue requests. Furthermore we show that as a nice side-effect such knowledge elicitation process helps to determine an exact, bidirectional, match between demand and supply.

We start by describing what is meant for a semantic-enabled marketplace. It is basically a repository, distributed or not, where advertisements, *i.e.*, demands and supplies, are fed or discovered, and both are expressed in a language which is unambiguous and has well-defined semantics. Usually this is done providing a shared ontology, describing the marketplace and its goods, written using languages, such as OWL-DL, which guarantee sound and complete reasoning services.

Available advertisements can then be matched with a request to find most promising ones. These are presented to the requester, a human or his/her agent, and a selection/negotiation process can follow. Systems that enable the matching process using the semantics of descriptions are called semantic matchmakers or facilitators [14, 20, 12, 7]. The basic issue in such facilitators is finding, given a request *i.e.*, a demand or a supply –depending on the perspective– satisfying, or at least promising, counterparts. Various match types may ensue, including partial, potential, full and exact (see [12] for a description of partial and potential classifications). An exact or complete match corresponds to the –rare– situation when an equivalence relation holds between a Demand D and a Supply S, *i.e.*, both the information in S imply those in D and those in D imply S. Full match corresponds to D implies S (or viceversa), potential to D and S being not contradictory in their specifications, partial to D and S presenting contradictory requirements.

If the equivalence does not hold and we are still looking for a match, it is necessary to hypothesize what is missing, or contradictory, on both sides. Such process requires inductive services typical of belief revision [15]. Also when a full match has been obtained, it can be the case, as shown in the introduction, that further interaction is needed. In this paper we argue that such information can be hypothesized eliciting knowledge from the marketplace, instead of requiring a domain knowledge that might be either irrelevant or hard to be fully determined. In this way a user is encouraged in learning, when needed, further information, without necessarily being endowed from the beginnning of a complete domain knowledge, and refine, in a principled way his/her request.

The remaining of the paper is structured as follows: next section revises Description Logics, which is the logic formalism we adopt here and inference services that will be used in the following. Then we briefly present our setting. In Section 4, we present and motivate algorithms for knowledge elicitation using the above mentioned inference services, also providing a simple example behavior. Discussion and conclusions close the paper.

2. BACKGROUND

Description Logics (DLs) are a family of logic formalisms for Knowledge Representation [1] that allow to model a domain of interest in terms of *concepts* –unary predicates– that describe subsets of objects in the domain and *roles* –binary predicates– that describe relations between objects. *Individuals* are used for special named elements belonging to concepts. Recently, motivated by the ongoing effort known as the Semantic Web initiative, DLs have been widely proposed in the modeling of e-marketplaces. [13, 22, 21, 11, 16, 12, 6]. DLs, in fact, allow for an open-world assumption, *i.e.*, incomplete information is admitted, and absence of information can be distinguished from negative information. Furthermore such languages allow to model constraints of structured descriptions as concepts, sharing a common ontology. Formally, concepts are interpreted as subsets of a domain of interpretation Δ, and roles as binary relations (subsets of $\Delta \times \Delta$). Basic elements can be combined using *constructors* to form concept and role *expressions*, and each DL has its distinguished set of constructors.

We give now a more formal definition of the above outlined basic elements by introducing the concept of semantic *interpretation*.

DEFINITION 1. **Semantic Interpretation**. A pair $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, made up of a *domain* Δ and an *interpretation function* $\cdot^{\mathcal{I}}$, which maps every concept to a subset of Δ, every role to a subset of $\Delta \times \Delta$, and every individual to an element of Δ.

Usually a so called *Unique Name Assumption*(UNA) is considered, which ensures different individuals to be mapped to different elements of Δ, *i.e.*, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for individuals $a \neq b$.

Every DL allows one to combine basic elements using *constructors* to form concept and role *expressions*. Each DL has its distinguished set of constructors, though all of them provide the *conjunction* of concepts, usually denoted as $\sqcap$. Among the distinguishing concept expressions constructors we enumerate disjunction $\sqcup$ of concepts and complement $\neg$ to close concept expressions under boolean operations.

Role expression can be obtained by combining roles with concepts using *existential role quantification* and *universal role quantification*. Other constructs may involve counting, as *number restrictions*.

Many other constructs can be defined, increasing the expressive power of the DL, up to n-ary relations [4]. Nevertheless, it is a well known result [2] that this usually leads to an explosion in computational complexity of inference services. Hence a trade-off is necessary. Once expressions have been built, they are given a semantics by defining the interpretation function over each construct.

Concept expressions can be used in *inclusion assertions*, and *definitions*, which impose restrictions on possible interpretations according to the knowledge elicited for a given domain. Definitions are useful to give a meaningful name to particular combinations. Sets of such inclusions are called TBox (Terminological Box). Individuals can be asserted to belong to a concept using membership assertions in an ABox. The semantics of inclusions and definitions is based on set containment: an interpretation $\mathcal{I}$ satisfies an inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and it satisfies a definition $C = D$ when $C^{\mathcal{I}} = D^{\mathcal{I}}$. A *model* of a TBox $\mathcal{T}$ is an interpretation satisfying all inclusions and definitions of $\mathcal{T}$. DL-based systems are equipped with reasoning services: logical problems whose solution can make explicit knowledge that was implicit in the assertions.

Basic reasoning services for $\mathcal{T}$, which are available in every DL-system are:

Concept Satisfiability: given a TBox $\mathcal{T}$ and a concept C, does there exist at least one model of $\mathcal{T}$ assigning a non-empty extension to C?

Subsumption: given a TBox $\mathcal{T}$ and two concepts C and D, is C more general than D in any model of $\mathcal{T}$?

Both Subsumption and Concept Satisfiability are adequate in all those scenarios where a yes/no answer is enough. For example, given a supply and a demand represented respectively by a concept $\mathcal{S}$ and a concept $\mathcal{D}$, using Concept Satisfiability we are able to establish whether they are compatible,*i.e.*, $\mathcal{S}$ models information which is not in conflict with the ones modeled by $\mathcal{D}$. This task can be performed checking the satisfiability of the concept $\mathcal{S} \sqcap \mathcal{D}$.
On the other hand Subsumption can be used to verify, for example, if a supply described by $\mathcal{S}$ satisfies a demand $\mathcal{D}$. It is easy understandable that if the relation $\mathcal{S} \sqsubseteq \mathcal{D}$ holds, then $\mathcal{S}$ is more specific than $\mathcal{D}$ and contains at least all the requested features.

Other inference services are needed when a binary response can be not enough. Let us consider concepts $\mathcal{S}$and $\mathcal{D}$, if their conjunction $\mathcal{S} \sqcap \mathcal{D}$ is unsatisfiable in the TBox $\mathcal{T}$ representing the ontology, *i.e.*, they are not compatible with each other, we may want to retract requirements in $\mathcal{D}$, G (for *Give up*), to obtain a concept K (for *Keep*) such that $K \sqcap \mathcal{S}$ is satisfiable in $\mathcal{T}$.

DEFINITION 2. *Let $\mathcal{L}$ be a DL, $\mathcal{S}$, $\mathcal{D}$, be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both $\mathcal{S}$ and $\mathcal{D}$ are satisfiable in $\mathcal{T}$. A* Concept Contraction Problem *(CCP) [5], identified by $\langle \mathcal{L}, \mathcal{D}, \mathcal{S}, \mathcal{T} \rangle$, is finding a pair of concepts $\langle G, K \rangle \in \mathcal{L} \times \mathcal{L}$ such that $\mathcal{T} \models \mathcal{D} \equiv G \sqcap K$, and $K \sqcap \mathcal{S}$ is satisfiable in $\mathcal{T}$. We call K a* contraction *of $\mathcal{D}$ according to $\mathcal{S}$ and $\mathcal{T}$.*

We use $\mathcal{Q}$ as a symbol for a CCP, and we denote with $SOLCCP(\mathcal{Q})$ the set of all solutions to a CCP $\mathcal{Q}$. We note that there is always the trivial solution $\langle G, K \rangle = \langle \mathcal{D}, \top \rangle$ to a CCP. This solution corresponds to the most drastic contraction, that gives up everything of $\mathcal{D}$. In a marketplace, it models the situation in which, in front of some very appealing supply $\mathcal{S}$, incompatible with the requested one, a user just gives up completely his/her specifications $\mathcal{D}$ in order to meet $\mathcal{S}$. On the other hand, when $\mathcal{S} \sqcap \mathcal{D}$ is satisfiable in $\mathcal{T}$, the "best" possible solution is $\langle \top, \mathcal{D} \rangle$, that is, give up nothing — if possible. Hence, a Concept Contraction problem is an extension of a satisfiability one. Since usually one wants to give up as few things as possible, some minimality in the contraction must be defined [15]. In most cases a pure logic-based approach could be not sufficient to decide between which beliefs to give up and which to keep. There is the need of modeling and defining some extra-logical information that have to be taken into account. One approach is to give up minimal information [5]. Another one considers some information more important than other and the piece of information that should be retracted is the least important one, that is negotiable and strict constraints are introduced [6].

If the offered supply $\mathcal{S}$ and the requested one $\mathcal{D}$ are compatible with each other, the partial specifications problem still holds, that is, it could be the case that $\mathcal{S}$ — though compatible — does not imply $\mathcal{D}$. Then, it is necessary to assess what should be hypothesized (H) in $\mathcal{S}$ in order to completely satisfy $\mathcal{D}$.

DEFINITION 3. *Let $\mathcal{L}$ be a DL, $\mathcal{S}$, $\mathcal{D}$, be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both $\mathcal{S}$ and $\mathcal{D}$ are satisfiable in $\mathcal{T}$. A* Concept Abduction Problem *(CAP) [10], identified by $\langle \mathcal{L}, \mathcal{D}, \mathcal{S}, \mathcal{T} \rangle$, is finding a concept $H \in \mathcal{L}$ such that $\mathcal{T} \models \mathcal{S} \sqcap H \sqsubseteq \mathcal{D}$, and moreover $\mathcal{S} \sqcap H$ is satisfiable in $\mathcal{T}$. We call H a* hypothesis *about $\mathcal{S}$ according to D and $\mathcal{T}$.*

We use $\mathcal{P}$ as a symbol for a CAP, and we denote with $SOL(\mathcal{P})$ the set of all solutions to a CAP $\mathcal{P}$. Observe that in the definition, we limit to satisfiable $\mathcal{S}$ and $\mathcal{D}$, since $\mathcal{D}$ unsatisfiable implies that the CAP has no solution at all, while $\mathcal{S}$ unsatisfiable leads to counterintuitive results ($\neg\mathcal{D}$ would be a solution in that case). If $\mathcal{S} \sqsubseteq \mathcal{D}$ then we have $H = \top$ as a solution to the related CAP. Hence, Concept Abduction extends subsumption. On the other hand, if $\mathcal{S} \equiv \top$ then $H \sqsubseteq \mathcal{D}$.

$\mathtt{SingleRoom} \equiv \mathtt{Bedroom} \sqcap (\geq 1\ \mathtt{hasBed})$
$\mathtt{DoubleRoom} \equiv \mathtt{Bedroom} \sqcap (\geq 1\ \mathtt{hasBed}) \sqcap \forall \mathtt{hasBed}.(\geq 2\ \mathtt{hasPlace})$
$\mathtt{WeddingRoom} \sqsubseteq \mathtt{DoubleRoom} \sqcap (\leq 1\ \mathtt{hasBed})$
$\mathtt{RoomFacility} \sqsubseteq \mathtt{Facility}$
$\mathtt{InternetAccess} \sqsubseteq \mathtt{RoomFacility}$
$\mathtt{WirelessInternet} \sqsubseteq \mathtt{InternetAccess}$
$\mathtt{SPA} \sqsubseteq \mathtt{Facility}$
$\mathtt{SwimmingPool} \sqsubseteq \mathtt{Facility}$
$\mathtt{Breakfast} \sqsubseteq \mathtt{Facility}$
$\mathtt{SmokingAllowed} \sqsubseteq \neg\mathtt{NoSmoking}$

Figure 1: The simplified ontology used as reference in examples

Notice that both Concept Abduction and Concept Contraction can be used for providing respectively subsumption and satisfiability explanation hypothesis. For Concept Contraction , having two concepts not compatible with each other, in the solution $\langle G, K \rangle$ to the CCP $\langle \mathcal{L}, \mathcal{D}, \mathcal{S}, \mathcal{T} \rangle$, G represents "why" $\mathcal{D} \sqcap \mathcal{S}$ are not compatible. For Concept Abduction , having $\mathcal{D}$ and $\mathcal{S}$ such that $\mathcal{S} \not\sqsubseteq \mathcal{D}$, the solution H to the CAP $\langle \mathcal{L}, \mathcal{D}, \mathcal{S}, \mathcal{T} \rangle$ represents "why" the subsumption relation does not hold. H can be interpreted as *what is specified in $\mathcal{D}$ and not in $\mathcal{S}$.*

3. THE E-MARKETPLACE

Current languages for describing products and services, such as UNSPSC, ECLASS, EOTD, lack of the needed coverage of concepts and of semantically rich and precise descriptions. It is doubtful they will be used as they are in Semantic Web oriented marketplaces. We envisage a marketplace, populated with semantic-endowed advertisements as proposed in [6], where users –or their agents– issue their requests of products and services as natural language advertisements, which are then automatically translated in structured DL concepts, using approaches such as the ones recently proposed in [17, 9, 18]. This appears necessary if we expect common users to practically use such systems, as it does not appear feasible that users have to become both logic and domain experts to post advertisements and to submit queries. In this realistic perspective it is reasonable that users may want to use their limited domain knowledge to interact with the system.

We also note that we assume that a common ontology is shared for advertisements; obviously if this does not hold true, an integration is required, as proposed *e.g.*, in [3]. It is noteworthy that, when schema integration techniques are used, it will be more probable that each user be endowed only of a partial knowledge of the domain, and the need for automated knowledge elicitation will be even more likely to emerge.

To simplify illustration of examples, we present in figure 1 a simplified reference ontology, which will be used in what follows. Notice that, but for examples, the approach proposed here is not bound to a particular DL, as we only require the DL be endowed of the conjunction minimal solution.

4. MANAGING BONUSES

As stated in the Introduction, in order to reach an equivalence relation between a demand and a supply, hypothesis have to be made both for supply refinement specification and in the demander knowledge of the marketplace domain. In this scenario, two kind of actors are represented. An active actor A and a passive one P. The former is able to actively retract part of his/her dynamic request/offer in order to be compatible with the latter. This one has a static representation of his offer/request. In DLs terms, if $A \sqcap P \equiv \bot$, then a contraction on A is performed, $A = \langle G_A, K_A \rangle$, and the matchmaking process continues considering only K_A, that is the part of A which is compatible with P; by definition $K_A \sqcap \mathcal{P} \not\equiv \bot$ (for further details see [6]). During the search for best possible matches within the marketplace, the active and passive actor role cannot change. Without loss of generality, for what concerns the active actor, a representation and management of negotiable and strict constraints is also possible, as proposed in [6]. In [6] also methodologies and measures able to estimate how good is the satisfaction degree of the demand, with respect to a minimal set of hypotheses in the supply, are provided and investigated. Such investigation can be brought back to describe "how well" the supply implies the demand. Nevertheless, nothing is said there about the converse implication.

At this point a further insight on the meaning of such implication is necessary with respect to an open world semantics scenario. Actually, if the supplier specifies information on a good characteristic or usage which is not in the user request, this information is not used in the matchmaking process. That is, the bonuses offered by the supplier have no weight while retrieving appealing supplies for the demander. On the other hand, if in the supply there is no bonus, it means that the information modeling the demand implies the supply one. A set-based example should clarify the latter observation. Suppose to have a request
$A = \{\texttt{Bedroom}, \texttt{SmokingAllowed}, \texttt{Breakfast}\}$
and a supply
$P = \{\texttt{Bedroom}, \texttt{InternetAccess}, \texttt{SPA}\}$.
With respect to A, we obtain that the bonus set is
$\mathcal{B} = \{\texttt{InternetAccess}, \texttt{SPA}\}$.
If such bonuses are canceled from P, *i.e.*, there are no more bonuses in P, the implication relation is reached.

$\{\texttt{Bedroom}, \texttt{SmokingAllowed}, \texttt{Breakfast}\} \supseteq P \backslash \mathcal{B}$
$= \{\texttt{Bedroom}\}$

Equivalently, the same relation holds if we add the bonuses to A.

$A \cup \mathcal{B} = \{\texttt{Bedroom}, \texttt{SmokingAllowed}, \texttt{Breakfast},$
$\texttt{InternetAccess}, \texttt{SPA}\} \supseteq P$

The first process, canceling bonuses from P, can be seen as a supply underspecification, the latter one as a query enrichment based on information which are elicited from P. In this case, bonuses can be seen as what has to be hypothesized in A in order to make P implied by A, which may lead to an actual exact match.

4.1 DLs Approach to Bonuses Management

Turning back to the DL-based scenario there is an active actor A able to contract his/her request –$A = \langle G_A, K_A \rangle$– if it is not compatible with a supply, and a passive actor unable to modify his supply (what is said in the following maintains its validity even if A is the supply and P is the demand). When an inconsistency between a demand and a supply ensues, the only way to conclude the matchmaking process is by contracting A and subsequently continuing using only K_A, that is the part of A which is compatible with P. At this point we extend the approach proposed in [6] considering bonuses in order to try reaching the equivalence relation between the demand and the supply.

Solving the Concept Abduction Problem $\langle \mathcal{L}, P, K_A, O \rangle$, where O represents the marketplace domain ontology, we obtain a solution H representing what has to be hypothesized and added to K_A in order to obtain $K_A \sqcap H \sqsubseteq P$. Hence, H, from now on B (for Bonus), represents the set of bonuses offered by P.

By definition it results both $K_A \sqcap H \not\equiv \bot$ and $A \sqcap H \equiv \bot$. Again a set-based example should be useful. Consider
$A = \{\texttt{Bedroom}, \texttt{SmokingAllowed}, \texttt{SPA}\}$
and
$P = \{\texttt{Bedroom}, \texttt{NoSmoking}, \texttt{Breakfast}\}$.
Due to the *(no)smoking* constraint, A and P are not compatible. Hence we rewrite $A = \{\texttt{SmokingAllowed}\} \cup \{\texttt{Bedroom}, \texttt{SPA}\}$ where, **in the sets world**, $G_A = \{\texttt{SmokingAllowed}\}$ and $K_A = \{\texttt{Bedroom}, \texttt{SPA}\}$. Now we have K_A and P compatible with each other. The bonus in P with respect to K_A are then
$B = \{\texttt{NoSmoking}, \texttt{Breakfast}\}$.
It is easy to notice that
$A \cup B = \{\texttt{Bedroom}, \texttt{SmokingAllowed}, \texttt{SPA},$
$\texttt{NoSmoking}, \texttt{Breakfast}\}$
is not consistent because we refine the request asking for both $\texttt{SmokingAllowed}$ and $\texttt{NoSmoking}$.

The latter problem arises because in P there is still the source of inconsistency with A, in this case $\texttt{NoSmoking}$. Then a contraction on P is also needed. In fact contracting P with respect to A results $P = \{\texttt{NoSmoking}\} \cup \{\texttt{Breakfast}, \texttt{Bedroom}\}$ where $G_P = \{\texttt{NoSmoking}\}$ and $K_P = \{\texttt{Breakfast}, \texttt{Bedroom}\}$. Hence, the bonuses looked for in K_P with respect to K_A are $B = \{\texttt{Breakfast}\}$.

Trivially, all the domain knowledge can be hypothesized leading the demand information to imply the supply one. Since one wants to model Ockhams razor[1], some minimality in the hypotheses must be defined.

In [10], among others, the conjunction minimal solution to a CAP is proposed for DLs admitting a normal form with conjunctions of concepts. A solution belonging to such solution is in the form $B = \sqcap_{i=1..k} b_i$, where b_i are DLs concepts and and is **irreducible**, *i.e.*, B is such that for each $h \in 1, ..., k$, $\sqcap_{i=1..h-1,h+1..k} b_i$ is not a solution for the CAP.

In the following the algorithm $computeExact(\mathcal{R}, A, O)$ is presented. It takes as input a set of resources $\mathcal{R} = \{P_1, ..., P_n\}$, a request A and the ontology with respect to which they all are modeled.

Algorithm $computeExact(\mathcal{R}, A, O)$
input DL concepts $P_i \in \mathcal{R}$, A, O reference ontology
output $\mathcal{B}_{irr}$ a set of DL concepts representing bonuses
begin algorithm
1: $\mathcal{B} = \emptyset$;
2: $\mathcal{B}_{irr} = \emptyset$;
3: **for each** $P_i \in \mathcal{R}\{$
4: $\langle G_A, K_A \rangle = contract(P_i, A, O)$;

[1] "entia non sunt multiplicanda prater necessitatem"

5: $\langle G_{P_i}, K_{P_i}\rangle = contract(A, P_i, O)$;
6: $B_i = abduce(K_A, K_{P_i}, O)$;
7: $\mathcal{B} = \mathcal{B} \cup \{\langle B_i, P_i\rangle\}$;
8: }
9: **for each** $\langle B_i, P_i\rangle \in \mathcal{B}$
10: **for each** $b_j \in B_i$
11: $\mathcal{B}_{irr} = \mathcal{B}_{irr} \cup \{b_j\}$;
12: **return** $\langle \mathcal{B}, \mathcal{B}_{irr}\rangle$;
end algorithm

The returned set $\mathcal{B}_{irr}$ contains all the bonuses available in the resources, which can be used to refine the query, and what is still missing for each available resource to obtain an exact, bidirectional, match.

With $contract(P_i, A, O)$ a call is indicated to a function able to contract A with respect to each P_i, conversely for $contract(A, P_i, O)$. With $abduce(K_A, K_{P_i}, O)$ a call is indicated to a function able to solve the related Concept Abduction Problem. Notice that in rows 9: - 11: a conjunction minimal solution is considered.

4.2 Example behavior

We present here an example to better clarify the algorithm behavior. The Description Logic adopted here is $\mathcal{ALN}$ with acyclic simple-TBox[1], which satisfies the condition for the conjunction minimal solution.

Let us consider the following set $\mathcal{R} = \{P_1, P_2, P_3\}$, related to the hotel booking scenario presented in Section 1, whose domain ontology is represented in Figure 1:

P_1. *Double room, Smoking permitted. Hotel have an excellent swimming pool available for the exclusive use of their guests.*
$P_1 = \mathtt{DoubleRoom} \sqcap \exists\mathtt{withFacility} \sqcap \forall\mathtt{withFacility}.(\mathtt{SmokingAllowed} \sqcap \mathtt{SwimmingPool})$

P_2. *Wedding room in an hotel four stars. Smoking is not permitted, price includes breakfast and use of SPA.*

$P_2 = \mathtt{WeddingRoom} \sqcap \exists\mathtt{withFacility} \sqcap \forall\mathtt{withFacility}.\mathtt{NoSmoking} \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.(\mathtt{Breakfast} \sqcap \mathtt{SPA}) \sqcap \exists\mathtt{class} \sqcap \forall\mathtt{class}.(= 4\ \mathtt{stars})$

P_3. *Wedding room in a three stars hotel . The price includes breakfast.*
$P_3 = \mathtt{WeddingRoom} \sqcap \exists\mathtt{class} \sqcap \forall\mathtt{class}.(= 3\ \mathtt{stars})$
$\exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.\mathtt{Breakfast}$

Eugenio's request is:

A. *I'm looking for a bedroom for two persons in an hotel at least three stars. Use of SPA included.*
$A = \mathtt{Bedroom} \sqcap \forall\mathtt{hasBed}.(= 2\ \mathtt{hasPlace}) \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.\mathtt{SPA} \sqcap \exists\mathtt{class} \sqcap \forall\mathtt{class}.(\geq 4\ \mathtt{stars})$

Looking for matches between A and both P_1 and P_2 we see that:

- $A \sqcap P_1 \not\equiv \bot$, then no contraction is needed and P_1 is a potential match for A, since P_1 fulfills partially the Eugenio's request.
- $A \sqcap P_2 \not\equiv \bot$, then also here no contraction is needed and P_2 seems to be the best match for A (see [6]).
- $A \sqcap P_3 \equiv \bot$, that is, A and P_3 are not compatible with each other. Hence a contraction on A is needed in order to continue the matchmaking process.

Based on the above considerations, after the matchmaking process the ranked list w.r.t. A is (P_2, P_1, P_3). In the following the result is presented obtained computing the bonus using $computeExact(\mathcal{R}, A, O)$ indicating each step within the algorithm rows.

4: $\langle G_A, K_A\rangle = \langle \top\ , \mathtt{Bedroom} \sqcap \forall\mathtt{hasBed}.(= 2\ \mathtt{hasPlace}) \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.\mathtt{SPA} \sqcap \exists\mathtt{class} \sqcap \forall\mathtt{class}.(\geq 4\ \mathtt{stars})\rangle$

5: $\langle G_{P_1}, K_{P_1}\rangle = \langle \top\ , \mathtt{DoubleRoom} \sqcap \exists\mathtt{withFacility} \sqcap \forall\mathtt{withFacility}.(\mathtt{SmokingAllowed} \sqcap \mathtt{SwimmingPool})\rangle$

6: $B_1 = (\geq 1\,\mathtt{withFacility}) \sqcap \forall\mathtt{withFacility}.(\mathtt{SmokingAllowed} \sqcap \mathtt{SwimmingPool})$

7: $\mathcal{B} = \{\langle B_1, P_1\rangle\}$

4: $\langle G_A, K_A\rangle = \langle \top\ , \mathtt{Bedroom} \sqcap \forall\mathtt{hasBed}.(= 2\ \mathtt{hasPlace}) \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.\mathtt{SPA} \sqcap \exists\mathtt{class} \sqcap \forall\mathtt{class}.(\geq 4\ \mathtt{stars})\rangle$

5: $\langle G_{P_2}, K_{P_2}\rangle = \langle \top\ , \mathtt{WeddingRoom} \sqcap \exists\mathtt{withFacility} \sqcap \forall\mathtt{withFacility}.\mathtt{NoSmoking} \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.(\mathtt{Breakfast} \sqcap \mathtt{SPA}) \sqcap \exists\mathtt{class} \sqcap \forall\mathtt{class}.(= 4\ \mathtt{stars})\rangle$

6: $B_2 = (\geq 1\ \mathtt{withFacility}) \sqcap \forall\mathtt{withFacility}.\mathtt{NoSmoking} \sqcap \forall\mathtt{priceIncludes}.\mathtt{Breakfast}$

7: $\mathcal{B} = \{\langle B_1, P_1\rangle, \langle B_2, P_2\rangle\}$

4: $\langle G_A, K_A\rangle = \langle \forall\mathtt{class}.(\geq 4\ \mathtt{stars})\ , \mathtt{Bedroom} \sqcap \forall\mathtt{hasBed}.(= 2\ \mathtt{hasPlace}) \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.\mathtt{SPA} \sqcap \exists\mathtt{class}\rangle$

5: $\langle G_{P_3}, K_{P_3}\rangle = \langle \forall\mathtt{class}.(= 3\ \mathtt{stars})\ , \mathtt{WeddingRoom} \sqcap \exists\mathtt{class} \sqcap \exists\mathtt{priceIncludes} \sqcap \forall\mathtt{priceIncludes}.\mathtt{Breakfast}\rangle$

6: $B_3 = \forall\mathtt{priceIncludes}.\mathtt{Breakfast}$

7: $\mathcal{B} = \{\langle B_1, P_1\rangle, \langle B_2, P_2\rangle, \langle B_3, P_3\rangle\}$

9-11: $\mathcal{B}_{irr} = \{(\geq 1\,\mathtt{withFacility}), \forall\mathtt{withFacility}.\mathtt{NoSmoking}, \forall\mathtt{priceIncludes}.\mathtt{Breakfast}, \forall\mathtt{withFacility}.(\mathtt{SmokingAllowed} \sqcap \mathtt{SwimmingPool})\}$

At this point Eugenio is able to refine and enrich his query adding some of the new information suggested in $\mathcal{B}_{irr}$, changing the rank in the matchmaking list.

5. DISCUSSION AND CONCLUSION

Query refinement methods [19] have been introduced, several years ago, in the framework of unstructured information retrieval (IR), usually textual IR, more recently also Multimedia IR.

There the basic idea is to present the user with a ranked list of resources (*e.g.*, documents, images, video fragments) the system has retrieved based on the original user query. The user becomes an active part of the loop in the successive retrieval stages by selecting one or more resources he/she judges relevant/irrelevant. In the well known Vector Space Model [19] this amounts to determine a new query properly weighing features of selected resources –thus performing an implicit knowledge elicitation– in the hope that the new query better corresponds to the user information need.

In our perspective the system elicites new knowledge to the user –what we call bonuses– extracting it from the marketplace available descriptions. This new system provided knowledge is submitted to the user, who may then decide to use new information or discard them, in further stages, thus enriching and refining the request. This process provides the user of an e-marketplace with novel information, without the burden of having *a priori* a thorough domain knowledge. The approach is along the line of explanation hypothesis, and helps to determine an exact match between demand and supply.

With reference to classic knowledge elicitation techniques the interested reader is referred to a recent survey on the topic [8]. Here we limited our interest to the elicitation of new knowledge from the descriptions, exploiting inference services, typical of belief revision, in a DL sematic-enabled marketplace, and motivated how this approach can help in simplifying interaction with users while improving requests precision.

Acknowledgements

We acknowledge partial support of projects CNOSSO, MS3DI, and PITAGORA.

6. REFERENCES

[1] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2002.

[2] R. Brachman and H. Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI-84)*, pages 34–37. Morgan Kaufmann, Los Altos, 1984.

[3] A. Calì, D. Calvanese, G. D. Giacomo, and M.Lenzerini. Data integration under integrity constraints. *Information Systems*, 29(2):147–163, 2004.

[4] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the Decidability of Query Containment under Constraints. In *Proceedings of the Seventeenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[5] S. Colucci, T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello. Concept Abduction and Contraction in Description Logics. In *Proceedings of the 16th International Workshop on Description Logics (DL'03)*, volume 81 of *CEUR Workshop Proceedings*, September 2003.

[6] S. Colucci, T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello. Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. In *Proceedings of the 6th Intl. Conf. on Electronic Commerce, ICEC'04*, pages 41–50, October 2004.

[7] S. Colucci, T. D. Noia, E. D. Sciascio, F. Donini, and M. Mongiello. Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. *Electronic Commerce Research and Applications*, 4(3), 2005.

[8] N. J. Cooke. Knowledge elicitation. In *Handbook of Applied Cognition*, pages 479–509. Wiley, 1999.

[9] S. Coppi, T. D. Noia, E. Di Sciascio, F. Donini, and A. Pinto. Ontology-based natural language parser for e-marketplaces. In *18th Intl. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, volume 3533 of *Lecture Notes in Artificial Intelligence*, pages 279–289. Springer-Verlag, 2005.

[10] T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello. Abductive matchmaking using description logics. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 337–342, Acapulco, Messico, August 9–15 2003. Morgan Kaufmann, Los Altos.

[11] T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello. Semantic matchmaking in a P-2-P electronic marketplace. In *Proc. Symposium on Applied Computing (SAC '03)*, pages 582–586. ACM, 2003.

[12] 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*, 8(4):9–37, 2004.

[13] E. Di Sciascio, F. Donini, M. Mongiello, and G. Piscitelli. A Knowledge-Based System for Person-to-Person E-Commerce. In *Proceedings of the KI-2001 Workshop on Applications of Description Logics (ADL-2001)*, volume 44 of *CEUR Workshop Proceedings*, 2001.

[14] M. Dumas, B. Benatallah, N. Russell, and M. Spork. A Configurable Matchmaking Framework for Electronic Marketplaces. *Electronic Commerce Research and Appplications*, 3(1):95–106, 2004.

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

[16] L. Li and I. Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. *International Journal of Electronic Commerce*, 8(4), 2004.

[17] V. Lopez and E.Motta. Ontology-driven question answering in aqualog. In *Proc. NLDB-04*, 2004.

[18] V. Lopez, M. Pasin, and E.Motta. Aqualog: An ontology-portable question answering system for the semantic web. In *The Semantic Web: Research and Applications, 2nd European Semantic Web Conf. ESWC '05*, volume 3532 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[19] G. Salton and M. M. Gill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[20] K. Sycara, M. Paolucci, M. Van Velsen, and J. Giampapa. The RETSINA MAS infrastructure. *Autonomous agents and multi-agent systems*, 7:29–48, 2003.

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

[22] D. Trastour, C. Bartolini, and C. Priest. Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. In *Proc. International World Wide Web Conference (WWW) '02*, pages 89–98. ACM, 2002.