Abductive Matchmaking using Description Logics

Tommaso Di Noia[(1)], Eugenio Di Sciascio[(1)], Francesco M. Donini[(2)] Marina Mongiello[(1)]
[(1)]Politecnico di Bari, Via Re David, 200, 70125 Bari, Italy
[(2)]Universita della Tuscia, Via S. Carlo 32, 01100 Viterbo, Italy
{t.dinoia,disciascio,mongiello}@poliba.it;donini@unitus.it

Abstract

Motivated by the matchmaking problem in electronic marketplaces, we study abduction in Description Logics. We devise suitable definitions of the problem, and show how they can model commonsense reasoning usually employed in analyzing classified announcements having a standardized terminology. We then describe a system partially implementing these ideas, and present a simple experiment, which shows the correspondence between the system behavior with human users judgement.

1 Motivation

We describe several scenarios in which matchmaking [Di Noia et al., 2003; Sycara et a/., 2002; Di Sciascio et al., 2001; Trastour *et al.,* 2002] is necessary, starting with a human situation and moving to more automated ones.

Human matchmaking. Imagine you are looking for an apartment in London, UK. On a local newspaper, you published a classified ad, which we call *D* (for Demand), asking the following:

> Apt. required: Soho, 2 rooms, smoker, dog, garden, max 600/month. Phone xxx-yyyyy

Since no one phoned you, you open a copy of the same newspaper, and start filtering a long list of classified ads offering apartments to rent. You might encounter the following ad, that we call S_1 (for Supply):

> Apt. centre (Piccadilly), car place, fireplace, 2 rooms. Phone zzz-wwwww

Then you wonder if this offer fits your needs. The match between Soho and Piccadilly depends on some domain knowledge — actually, they are consistent, although Piccadilly does not imply Soho. Then *S*\ does not mention that dogs and smokers are allowed, but this does not necessarily mean that they are not (open-world assumption). It docs not mention a garden too, but there is a car place, which could be turned into a small garden if it is open air and private. Should you phone to inquire *S*\ first, or phone first the following S_2:

> AAA Apt. in London. Phone sss-ttttt

Here it is not just a problem of hypothetical reasoning, it is also a problem of giving some total order to offers that are compatible with yours, presumably trying to maximize the probability of finding a good offer in few phone calls - but wait: if these offers would match, why *they* didn't phone *you'!*

Software agents matchmaking. A similar scenario, but for the fact that agents may actively look for a matching counteroffer, instead of simply posting their offer. So they may actually call you...

Matchmaking in Electronic marketplaces. The same scenario, but with some sort of standardized language to describe supplies and demands, and maybe some centralized facilitator for matchmaking, *e.g.,* containing the necessary domain knowledge.

Service Discovery. Either for software agents or in e-marketplaces, the offers may describe a general service in some standardized language, such as the simple 128-bit string of Bluetooth, or the more structured DAML-S.

The third and fourth scenarios are more amenable to automated matchmaking, since they imply a standardized language and ontology. However, observe that also in the first and second scenarios, when supplies and demands are expressed in heterogeneous forms, matchmaking is not a mere schema/data integration problem. In fact, integration techniques [Madhavan *et al.,* 2001] may be employed to make heterogeneous supplies and demands comparable; but once they are reformulated in a comparable way, one is still left with true matchmaking problems: *i)* given a proposal, are there compatible counteroffers? *ii)* if there are many compatible counteroffers, which are the most promising ones (and why)?

2 Principles for matchmaking

We now discuss in detail various technical options for performing matchmaking, first trying to highlight limitations of non-logical approaches, and then discussing the general Knowledge Representation principles that a logical approach may yield.

First of all, we note that if supplies and demands are simple names or strings, the only possible match would be identity, resulting in an all-or-nothing approach to matchmaking. This is the approach taken by Bluetooth, whose Service Discovery Protocol tries to match Universally Unique Identifiers of

services and requests. Although effective for fixed technical domains, such an approach misses the fact that proposals usually have some sort of structure in them. Such a structure could be exploited in order to evaluate "interesting" inexact matches.

Vector-based techniques taken by classical Information Retrieval can be used, too, thus reverting matchmaking to similarity between weighted vectors of stemmed terms, as proposed in the COINS matchmaker [Kuokka and Harada, 1996] or in LARKS [Sycara *et al,* 2002]. Obviously lack of document structure in descriptions would make matching only probabilistic and strange situations may ensue, *e.g.,* the demand "Apt. with two Rooms in Soho pets allowed no smokers" would match the supply "Apt. with two Rooms in Soho, no pets, smokers allowed".

A further possibility is to use sets of words to describe the structure of supplies and demands. We consider these description as a particular case of logical description, and move on to Description Logics (DLs). Hence, from now on we suppose that supplies and demands are expressed in a DL *C,* equipped with a model-theoretic semantics. We suppose also that a common ontology for supplies and demands is established, as a TBox T in L. Now a match between a supply S and a demand *D* could be evaluated according to T. This framework ensures the first property that we would like to hold for matchmaking, namely, an open-world assumption.

Property 1 (Open-world descriptions) *The absence of a characteristic in the description of a proposal 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 issuer of the proposal.*

Moreover, if all constraints of a demand *D* were fulfilled by a supply S, but not vice versa, then S should be among the top ranked supplies in the list of potential partners of the demander, while *D* should not appear at the top in the list of potential partners of the supplier.

Property 2 (Non-symmetric evaluation) *A matchmaking system may give different evaluations to the match between a supply S and a demand D, depending on whether it is trying to match S with D, or D with S* — i.e., *depending on who is going to use this evaluation.*

There are three relations between concepts expressing supplies and demands, that we consider meaningful in matchmaking: implication, consistency, and inconsistency.

In the first case, $\mathcal{T} \models (D \sqsubseteq S)$, *i.e.,* constraints imposed by *D* imply those of S, and vice versa if $\mathcal{T} \models (S \sqsubseteq D)$. This relation extends the previous set-based inclusion to general concepts. If both $\mathcal{T} \models (D \sqsubseteq S)$ **and** $\mathcal{T} \models (S \sqsubseteq D)$, then *D* and *S* should be considered equivalent in T. This relation extends exact matching, making syntax differences irrelevant. In case of consistency, $D \sqcap S$ is satisfiable in T. Then, there is a *potential* match, in the sense that the constraints of neither proposal exclude the other. This relation has been highlighted also by other researchers [Trastour *et al.,* 2002]. However, that proposal lacks a *ranking* between different potential matches. In the third case, $D \sqcap S$ is unsatisfiable in T. Although also in this case a matching could be defined [Di Noia *et al.,* 2003], we do not treat this case here for lack of space. Hence, from now on we concentrate on potential match only. We now highlight some principles that — we believe — every ranking function should have in logical matchmaking.

First of all, if a logic is used to give some meaning to descriptions of supplies and demands, then proposals with the same meaning should have the same ranking, independently of their syntactic descriptions.

Definition 1 (Syntax independence in ranking) *A ranking of matches is* syntax independent *if for every pair of supplies S_1 and S_2, demand D, and ontology T, when S\ is logically equivalent to S_2 then S\ and S_2 have the same ranking for D, and the same holds also for every pair of logically equivalent demands D\. D_2 with respect to every supply S.*

For example, an apartment *S\,* described as available for the summer quarter, should have the same rank — with respect to a request — as another S_2, identical but for the fact that it is described to be available for june-july-august. Clearly, when the logic admits a normal form of expressions — as CNF or DNF for propositional logic, or the normal form of concepts for the DL of CLASSIC [Borgida *et al.,* 1989] — using such a normal form ensures by itself syntax independence.

We now consider the relation between ranking and implications. We go back to the descriptions with sets of words, since they are easy to read through. Let *D* be a demand and S_1, S_2 be two supplies defined as follows:

D — {apartment, soho, two Rooms, pets Allowed)

S\ — {apartment, soho, boiler, quiet}

S_2 = *{apartment, soho, boiler, quiet, last Floor}*

In this case, the characteristics that S_2 adds to S_1 are irrelevant for *D.* Hence, whatever the rank for *S*, the one for S_2 should be the same. If instead we let

S_3 = *{apartment, soho, boiler, quiet, pets Allowed}*

then S3 should be ranked better than $_{S1}$ since it adds a characteristic required by *D.* We generalize this example to concepts, and state the following definition.

Definition 2 (Monotonicity of ranking over subsumption) *A ranking of potential matches is* monotonic over subsumption *whenever for every demand D, for every pair of supplies S\ and S_2, and ontology T, if S_1 and S_2 are both potential matches for D, and $\mathcal{T} \models (S_2 \sqsubseteq S_1)$, then S_2 should be ranked either the same, or better than S_1, and the same should hold for every pair of demands D_1,D_2 with respect to a supply S.*

Intuitively, the above definition could be read of as *A ranking is monotonic over subsumption if: the more specific, the better.* Observe that we use the word "better" instead of using any symbol <, >. This is because some rankings may assume that "better" means "increasing" (towards either infinity, or 1) while others may assume "decreasing" (towards 0). Observe also that the above definition refers only to potential matches, *i.e.,* consistent matches — otherwise, $\perp$ would be the best

match (being the most specific concept), which is obviously not the case.

We remark that the properties and definitions we stated in this section are independent of the particular DL employed, or even the particular *logic* chosen. For instance, the same properties could be stated if propositional logic was used to describe supplies, demands and the ontology. In this respect, we believe that this section keeps its significance also if one chooses more expressive DLs such as *SHOQ (V)* [Horrocks and Sattler, 2001] or DAML [Paolucci *et ai,* 2002].

3 Abduction in Description Logics

We tend to follow the notation of [Riter and Gottlob, 1995] for propositional abduction whenever possible, and adapt it to our setting.

Definition 3 *Let C be a DL, C, D, be two concepts in C, and T be a set of axioms in C. A* Concept Abduction Problem *(CAP), denoted as (C, C, D, T), is finding a concept* $H \in C$ *such that* $\mathcal{T} \not\models C \sqcap H \equiv \bot$, *and* $\mathcal{T} \models C \sqcap H \sqsubseteq D$.

We use P as a symbol for a CAP, and we denote with *SOL(V)* the set of all solutions to a CAP *P.* Observe that if $C \sqcap D$ is unsatisfiable in T, then trivially $SOL(\mathcal{P}) = \emptyset$, i.e.., there are no solution to *P* if *C* and *D* are not a potential match. Hence from now on we concentrate on the case $\mathcal{T} \not\models C \sqcap D \equiv \bot$.

As propositional abduction extends implication, a CAP extends concept subsumption. But differently from propositional abduction, we do not make any distinction between manifestations and hypotheses, which is usual when abduction is used for diagnosis. However, diagnosis is not our domain of application for abduction, and when making hypotheses about properties of goods in e-marketplaces, there is no point in making such a distinction. This uniformity implies that if $C \sqcap D$ is satisfiable in *T,* then $D \in$ *SOL((C,C, D,T)), i.e.,* there is always the trivial solution *D* to a non-trivial CAP (£, C, *D,T).* Interpreted in our e~ marketplace application domain, it means that if 1 hypothesize for the counteroffer exactly all my specifications, the counteroffer trivially meets my specifications. Since one wants to model Occam's razor, some minimality in the hypotheses must be defined. In the following definition, we denote with $\sqsubseteq_{\mathcal{T}}$ the subsumption relation between concepts w.r.t. a TBox T.

Definition **4** *Let* $\mathcal{P} = (\mathcal{L}, C, D, \mathcal{T})$ *be a CAP. The set* $SOL_{\sqsubseteq}(\mathcal{P})$ *is the subset of* $SOL(\mathcal{P})$ *whose concepts are maximal under* $\sqsubseteq_{\mathcal{T}}$. *The set* $SOL_{\leq}(\mathcal{P})$ *is the subset ofSOL(P) whose concepts have minimum length.*

We note that being maximal under $\sqsubseteq_{\mathcal{T}}$ is still a minimality criterion, since it means that no unnecessary hypothesis is assumed. It can be proved that the two measures are incomparable.

Proposition 1 *There exists a CAP P such that the two sets* $SOL_{\sqsubseteq}(\mathcal{P})$ *and* $SOL_{\leq}(\mathcal{P})$ *are incomparable.*

Proof. It is sufficient to consider $D = A_1 \sqcap A_2 \sqcap A_3$, $C = A_1$ a $\mathcal{T} = \{B \sqsubseteq A_2 \sqcap A_3\}$. The logic is even propositional. Then $A_2 \sqcap A_3 \in SOL_{\sqsubseteq}(\langle \mathcal{L}, C, D, \mathcal{T}\rangle)$, $B \in SOL_{\leq}(\langle \mathcal{L}, C, D, \mathcal{T}\rangle)$, and neither solution is in the other set. □

The proof highlights that, although $\leq$-minimality could be preferable for conciseness, it is heavily dependent on T. For every concept $H \in SOL(P)$, it is sufficient to add the axiom $A \equiv H$ to get a $\leq$-minimal solution *A.*

A third minimality criterion is possible for DLs that admit a normal form as a conjunction of concepts, that is, every concept C in *C* can be rewritten as an equivalent concept $C_1 \sqcap \cdots \sqcap C_n$. This is the case for L= *ACM,* and for the DL of the CLASSIC KR system. We call such a normal form CNF, in analogy with propositional logic.

Definition 5 *Let* P=(L, C, *D, T) be a CAP in which C admits a CNF. The set* $SOL_{\sqcap}(\mathcal{P})$ *is the subset of SOL{P) whose concepts are minimal conjunctions,* i.e., /f $C \in SOL_{\sqcap}(\mathcal{P})$ *then no sub-conjunction of C is in SOL(P). We call such concepts* irreducible *solutions of P.*

It turns out that $\sqcap$-minimality subsumes both $\sqsubseteq_{\mathcal{T}}$-minimality and $\leq$-minimality. This is not a surprise, since $\sqcap$-minimality is a form of $\sqsubseteq_{\emptyset}$-minimality, *i.e.,* maximality for subsumption w.r.t. an empty TBox.

Proposition 2 *For every CAP P in which C admits a CNF, both* $SOL_{\sqsubseteq}(\mathcal{P})$ *and* $SOL_{\leq}(\mathcal{P})$ *are included in* $SOL_{\sqcap}(\mathcal{P})$.

Proof. If a concept *C* is not $\sqcap$-minimal, then it is not $\leq$-minimal, and the same for $\sqsubseteq_{\mathcal{T}}$. ⊠

3.1 Computational Complexity

Since Concept Abduction extends Concept Subsumption w.r.t. a TBox, complexity lower bounds of the latter problem carry over to decision problems related to a CAP.

Proposition 3 *Let P=(C,C,D,T) be a CAP If Concept Subsumption w.r.t. a TBox in C is a problem C-hard for a complexity*[1] *class C, then deciding whether a concept belongs to SOL(P) is both C-hard and co-C-hard.*

Proof. Hardness for *C* comes from the fact that *C* is subsumed by D in T if T $\in$ *SOL{P).* □

Hence, if *C* contains the DL *AC,* then deciding whether a concept belongs to *SOL(P)* is EXPTIME-hard [Donini, 2003] for a general TBox T, but it is PSPACE-hard if T contains only acyclic concept axioms [Calvanese, 1996].

Regarding upper bounds, a simple result can be derived from the fact that *D* is always a solution of the CAP *(C,C,D,T)* — although not always a minimal one. First of all, a total length-lexicographic order $\prec$ can be defined over concepts as follows: given two concepts C, $D \in C$, let $C \prec D$ if either $|C| < |D|$, or both $|C| = |D|$ and *C* is lexicographically before *D.* Based on this total order, an algorithm for finding a $\leq$-minimal solution of a CAP, using polynomial space relatively to an oracle for subsumption in *C,* is the following:

input: a CAP $\mathcal{P} = \langle \mathcal{L}, C, D, \mathcal{T}\rangle$, with $\mathcal{T} \not\models C \sqcap D \equiv \bot$
output: a concept in $SOL_{\leq}(\mathcal{P})$
variables: concept $x \in \mathcal{L}$
begin
  x : = T ;

while $|x| < |D|$ **do**
if $(\mathcal{T} \models C \sqcap x \sqsubseteq D$ **and** $\mathcal{T} \not\models C \sqcap x \sqsubseteq \bot)$
then return *x;*
x := next concept following *x* in $\prec$
endwhile;
return *D*
end.

The above algorithm uses polynomial space (considering the two calls to subsumption as an oracle) since it just tries all concepts with less symbols than D, and returns D if it does not find a shorter solution. Therefore, it provides an upper bound on the complexity of CAP, depending on the complexity class to which subsumption in *C* belongs to. Although this result does not directly lead to a practical algorithm, it puts an upper bound on the complexity of the problem, hence on the complexity of every optimal algorithm.

Theorem 1 *Let P=(C, C,* D, *T) be a CAP. If subsumption in C belongs to a complexity class C that is included in PSPACE, then finding a concept in* $SOL_{\leq}(\mathcal{P})$ *is a problem in PSPACE. Otherwise ifPSPACE is included in C, then finding a concept in* $SOL_{\leq}(\mathcal{P})$ *is a problem in C.*

Given that the problem of finding a solution cannot be simpler than the corresponding decision problem, we can conclude with some general results about $\leq$-minimal abduction.

Theorem 2 *Let V={C,C,D,T) be a CAP, with C a DL whose expressiveness is between AC and the DL containing concept constructors* $\sqcap, \sqcup, \neg, \exists R.C, \forall R.C$ *and role constructors* $\sqcup$, *role chain, transitive-reflexive roles, role identity, role inverse. Then finding a concept in* $SOL_{\leq}(\mathcal{P})$ *is a problem EXPTIME-complete when T is a general TBox.*

Proof. Hardness results for *AC* are in [Donini, 2003]. Membership result for the most expressive logic comes from converse-Propositional Dynamic Logic [Vardi and Wolper, 1986]. □

Hence, for a general TBox the best known algorithms require exponential time and also exponential space (unless one proves PSPACE = EXPTIME).

When the TBox is acyclic, complexity results for subsumption imply that finding a concept in $SOL_{\leq}(\mathcal{P})$ is a problem PSPACE-complete for DLs whose expressiveness is between $\mathcal{ALE}$ [Calvanese, 1996] and *ACC* [Baader and Hollunder, 1991], Even for the simplest logic $\mathcal{AL}$, the problem is co-NP-hard [Calvanese, 1996].

3.2 Irreducible solutions in *ACN*

In this section, we assume that *T* of a CAP *V*= (£, C, Z,, *T)* is always acyclic. Finding an irreducible solution is easier than finding a $\leq$-minimal or a $\sqsubseteq$-minimal solution, since a greedy approach can be used to minimize the set of conjuncts in the solution: starting from $C \sqcap D$, delete one redundant conjunct at a time from *D.* However, instead of starting from $C \sqcap D$, we adapt a structural subsumption algorithm [Borgida and Patel-Schneider, 1994] that collects all concepts that should be conjoined to *C* to be subsumed by *D.* The algorithm operates on concepts in normal form; roughly speaking, this form is obtained by expanding concept names with their definition in T, and then applying the following rewriting rules as much as possible:

$$\begin{aligned} \forall R.C \sqcap \forall R.D &\rightarrow \forall R.C \sqcap D \\ C \sqcap \bot &\rightarrow \bot \\ (\geq n\ R) \sqcap (\leq m\ R) &\rightarrow \bot \text{ if } n > m \\ \forall R.\bot &\rightarrow \forall R.\bot \sqcap (\leq 0\ R) \end{aligned}$$

In the following algorithm, we denote the fact that a concept *A* appears as a conjunct of a concept *C* with $A \in C$.

Algorithm *findIrred(V);*
i a CAP $\mathcal{P} = \langle \mathcal{L}, C, D, \mathcal{T} \rangle$, with $\mathcal{L}=\mathcal{ALN}$, acyclic $\mathcal{T}$ and $\mathcal{T} \not\models C \sqcap D \equiv \bot$
output: concept $H \in SOL_{\sqcap}(\mathcal{P})$
(where *H* = T means that *C* is already subsumed by *D)*
variables: concept H
begin
// := T;
for every concept name *y* in *D*
if *y* is not in *C*
then $H := H \sqcap y$;
for every concept $(\geq n\ R)$ in D
such that there is **no concept** $(\geq m\ R)$ in C with $m \geq n$
$H := H \sqcap (\geq n\ R)$;
for **every concept** $(\leq n\ R)$ in D
such **that there is no concept** $(\leq m\ R)$ in C with $m \leq n$
$H := H \sqcap (\leq n\ R)$;
for every concept $\forall R.E$ in D
if there exists $\forall R.F$ in C
then $H := H \sqcap findIrred(\langle \mathcal{L}, F, E, \mathcal{T} \rangle)$
else $H := H \sqcap findIrred(\langle \mathcal{L}, \top, E, \mathcal{T} \rangle)$;
/* now $H \in SOL(\mathcal{P})$, but it might be reducible */
for every concept H_i in H
if H **without** $H_i \in SOL(\mathcal{P})$
then delete H_1 from *H;* return *H;*
end.

It can be proved that the concept *H* returned by *find Irred ()* is indeed an irreducible solution of *V.* As for complexity, the expansion of the TBox in the construction of the normal form can lead to an exponential blow-up, as demonstrated by Nebel in [1990]. And anyway, a polynomial algorithm cannot be expected since subsumption in *AC* with an acyclic T is co-NP-hard [Calvanese, 1996]. However, in the cited paper Nebel argues that the expansion is exponential in the depth of the hierarchy T; if the depth of T is 0(log |T|), then the expansion is polynomial, and so is the above algorithm.

Theorem 3 *Let V=* (£,C,D,T) *be a CAP, with C=AC, and T an acyclic TBox whose depth is always bounded by* $O(\log |\mathcal{T}|)$. *Then finding an irreducible solution to V is a problem solvable in polynomial time.*

In order to rank the proposals in a marketplace according to how "near" they are to a given proposal D, we take the number of concept names in the irreducible solution returned by the above algorithm. Although this makes the rank depend on the algorithm, it can be easily proved that this definition is in accordance with the principles highlighted in Section 2. Moreover, the fact that a concept *H* is actually computed

makes it easy to devise an explanation facility, in case a user wants to know why a given proposal has been ranked before another. This transparency is crucial for our electronic business scenario, where we must give the user reasons to trust the system. In the next section, we show the results of an experiment comparing the rank obtained by our deployed system and the rank provided by some users in the same scenario.

4 Experiments

Using the highlighted properties as a formal specification we designed and implemented a prototype facilitator. The system embeds an adapted NeoClassic reasoner, the C++ implementation of the original Classic. The rationale of the choice of the Classic system, apart from the obviously useful availability of concrete datatypes and the possibility to extend its functionalities through test functions, is its polynomial time inference. The system receives a description of demand/supply. Then, the reasoner checks the description for consistency; if it fails, based on the reasoner output, the system provides an error message stating the error occurred. Otherwise the proper matchmaking process takes place, and the system returns a ranked set of matches. When no exact match is available, The user may investigate potential matches. In this process he/she is helped by the system ranking. We believe that degree of conformance of an automated matchmaking facilitator to users' perception is of extreme importance, especially in a setting as our own, which ranks and categorizes matches. To evaluate the ability of the system to meet users' expectations we set up a little experimental framework.

We selected all apartments-rental advertisements from the dedicated section of a local newspaper on Oct. 6th of last year. We subdivided them in two sets, *demands* (23 advertisements) and *supplies* (39 advertisements). It is noteworthy that, at least for the single-day pick we made, we were not able to detect any exact match. We submitted to twenty volunteers of various sex and age, a questionnaire that included 8 items. Each item was one demand and a set of up to eight supplies, or one supply and a set of up to eight demands. Volunteers were asked to rank, according to their judgement, elements of each set with respect to the given advertisement, with the following question: Order the following Demands(Supply) with respect to the given Supply (Demands) as you would contact them had you issued the given Supply (Demand). Volunteers were given unlimited time and in average it took approximately half hour to complete the questionnaire. Then the same sets of items were submitted to the reasoner. As a general consideration, the response of the reasoner was quite close to the users' ones, and considering average volunteers orderings the system ranking was in agreement with the human judgement almost always. Average percentage of deviation from the mean values determined by the users on the whole experiment was 8.3.

As an example, we show here in more detail results of the evaluation for a single demand/supplies matching process, extracted form the experimental setting. Let us consider the example demand: *student looking for a nice 1/2 bed flat, ch, furnished, kitchen, washing machine. Price: 150*

Also consider the following advertisements submitted as supplies:

- *Supply 1: large, fully furnished room, price includes cable TV and bills. Price: 120*
- *Supply2: double room, suit a couple or two girls, required deposit. Price: 150*
- *Supply3: room to rent, suit a non smoking female student with worker, international preferred, deposit required. Price: 80*
- *Supply4: single room in clean flat for nosmoker quiet student, sharing with 2 others. Price: 120*
- *Supply5: dbl room in shared house, suit single person, use of lounge, kitchen , garden, rent includes council tax. Price: 85*
- *Supply6: female to share a room in a residential areaflatshare, washing machine, TV, VCR. Price: 81*
- *Supply 7: large room in family houseshare with 2 adults, suit prof / student female, viewing recommended. Price: 95*
- *Supply8: 2 bed flat, perfect for student, dble bed ADSL computer beneath. Price: 600*

```
demand (and Bedroom (all toLetFor Student) (at-leaBt 1 hasBed)
(at-most 2 hasBed) (at-least 1 hasFacilities) (all hasFacililies
(and WashingMachine NoAutonomousHeating FullyFurmshed)) (all
haaServices Kitchen) (all price (maximum 150)))

supplyl (and Bedroom (all price (minimum 120)) (all priceIncludes
(and Bill TVPrice)) (all hasFacilitieB (and FullyFurniBhed
Spatious)))

supply2 (and (all price (minimum 150)) (at-least 2 hasRoom) (at-mosr
2 hasRoom) (all hasRoom Room) (at-least 2 toLetFor) (at-tnoBt 2
toLetFor)(all toLetFor (and Couple Student))(at-least 1
depoaitRequired)(all depooitRequired Yes))

supply3 (and (at-least 1 depoaitRequired)(all deposit Required
Yes)(all price (minimum 80)) Bedroom (all toLetFor (and Student
NoSmoker Worker (all sex Female))))

supply4 (and SingleRoom (all toLetFor (and NoSmoker
Student))(at-leaat 2 occupants)(all price (minimum 120)))

Bupply5 (and DoubleRoom (all toLetFor Single)(all hasFacilities
(and Lounge Garden))(all hasServices Kitchen)(all price (minimum
85))(all priceIncludes CouncilTax))

supply6 (and Bedroom (at-least 1 occupants) (all price (minimum
81)) (all toLetFor Female)(ail hasFacilities (and WashingMachine
TV VCR)))

supply7 (and Bedroom (all price (minimum 95)) (all occupants
Family) (at-least 2 occupants) (all toLetFor (and Student
Professional (all sex Female))))

supplyB (and Flat (at-least 2 hasBed) (at-most 2 hasBed) (all
toLetFor Student) (all hasFacilities ADSL) (all price (minimum
9b))))
```

Figure 1: Neoclassic descriptions of demand and supplies

The translation of these advertisements into Classic, in accordance with our ontology is pictured in Figure 1. Table 1 shows results provided from the matchmaking algorithm comparing the system provided ranking and the average volunteers orderings for demands w.r.t. to a supply. Figure 2 shows the same results in a graphical form. Notice that, while volunteers gave strictly ranked orderings, the system could also provide equal ranking for various matches. Polygons represent a range for the system total order of supplies, when there are equally ranked supplies. For example, Supply 3 and 7 are given the same rank by the system, so they could be equally put in fourth or fifth position when supplies are given a total order. Hence we consider correct system

matched pairs	system ranking	averaged users ranking
demand,supply1	2-3	2
demand,supply2	8	6
demand,supply3	4-5	4
demand,supply4	2-3	3
demand,supply5	1	1
demand,supply6	6	5
demand,supply7	4-5	8
demand,supply8	7	7

Table 1: System results and user preferences

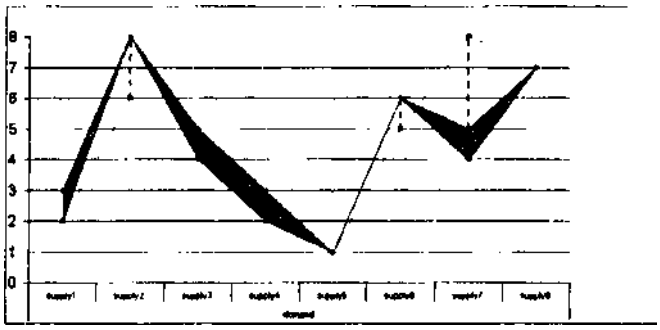

Figure 2: System and averaged users ranking

ranks for Supply 3 and 4, because users' rank fall inside the polygons, while we consider incorrect system placement of Supply 2 and 7 (dotted lines mark the difference).

5 Conclusion

Motivated by the matchmaking problem in electronic marketplaces, we have studied abduction in DLs. We have presented suitable definitions of the problem, and have shown how they can model commonsense reasoning usually employed in analyzing typical commercial advertisements. Although we used the simple DL *ACM,* our definitions and logical framework can be used also when a more expressive DL is used. Of course in this case more complex algorithms should be devised. To substantiate these ideas we have implemented a prototype based on the devised specifications adapting a NeoClassic reasoner, and presented experiments that show the correspondence between the system behavior with human users judgement.

Acknowledgements

We thank Peter Patel-Schneider for valuable suggestions on CLASSIC, Simona Colucci and Marco Mottola for useful implementations. This work has been supported by MURST project CLUSTER22, by EU-POP project *"Negotiation Agents for the Electronic Marketplace* ", by PON project *"Tecnologie innovative per la valorizzazione e la fruizione dei Beni Culturali",* by CNR projects LA1CO, and *"Metodi di Ragionamento Automatico nella modellazione ed analisi di dominio* ".

References

[Baader and Hollunder, 1991] F. Baader and B. Hollunder. K*RIS: K*nowledge Representation and Jnference System. *SIGART Bulletin,* 2(3):8-14, 1991.

[Borgida and Patel-Schneider, 1994] A. Borgida and P. F. Patel-Schneider. A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic../. *of Artif Intell Res.,* 1:277-308, 1994.

[Borgida et *al.,* 1989] A. Borgida, R. J. Brachman, D. L. McGuinness, and L. Alperin Resnick. CLASSIC: A structural data model for objects. In *ACM SIGMOD,* pages 59-67, 1989.

[Calvanese, 1996] D. Calvanese. Reasoning with inclusion axioms in description logics. In *ECAI'96,* pages 303-307. John Wiley & Sons, 1996.

[Di Noia *et al,* 2003] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Semantic matchmaking in a P-2-P electronic marketplace. In *SAC"03,* pages 582-586. ACM, 2003.

[Di Sciascio *etal,* 2001] E. Di Sciascio, F.M. Donini, M. Mongiello, and G. Piscitelli. A knowledge-based system for person-to-person e-commerce. In *ADL Workshop (KI2001),* 2001.

[Donini, 2003] F. M. Donini. Complexity of reasoning. In *Description Logics Handbook,* chapter 3. Cambridge University Press, 2003.

[Eiter and Gottlob, 1995] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *JACM,* 42(I):3-42, 1995.

[Horrocks and Sattler, 2001] I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *IJCAI200I,* pages 199-204, 2001.

[Kuokka and Harada, 1996] D. Kuokka and L. Harada. Integrating information via matchmaking. *J. of Intelligent Information Systems,* 6:261-279, 1996.

[Madhavan *et al.,* 2001] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proc. of VLDB* W, pages 49-58, 2001.

[Nebel, 1990] B. Nebel. Terminological reasoning is inherently intractable. *Artif Intell,* 43:235-249, 1990.

[Paolucci *etal.,* 2002] M. Paolucci, T. Kawamura, T.R. Payne, and K. Sycara. Semantic matching of web services capabilities. \n *ISWC 2002,* pages 333-347, 2002.

[Sycara *et al.,* 2002] K. Sycara, S. WidofY, M. Klusch, and J. Lu. Larks: Dynamic matchmaking among heterogeneus software agents in cyberspace. *Autonomous agents and multi-agent systems,* 5:173-203, 2002.

[Trastour *etal,* 2002] D. Trastour, C. Bartolini, and C. Priest. Semantic web support for the business-to-business e-commerce lifecycle. In *WWW02,* pages 89-98. ACM, 2002.

[Vardi and Wolper, 1986] M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *J. of Comp. and Syst. ScL,* 32:183-221, 1986.