Abduction and Contraction for Semantic-based Mobile Dating in P2P Environments

Michele Ruta, Tommaso Di Noia, Eugenio Di Sciascio, Floriano Scioscia
Politecnico di Bari
DEE-SisInfLab
via Re David 200, I-70125 Bari, Italy
{m.ruta, t.dinoia, disciascio, f.scioscia}@poliba.it

Abstract

In a generic semantic-based matchmaking process, given a request, it is desirable to obtain a ranked list of compatible services/resources/profiles in order of relevance. Furthermore, a match explanation can provide useful information to modify or refine the original request in a principled way. Though the feasibility of such an approach has been proved with fixed reasoning engines, it is a challenging subject to perform inference tasks on handheld devices. Here we propose abduction and contraction algorithms in Description Logics specifically devised for applications in mobile environments. A simple interaction paradigm based on Bluetooth protocol stack has also been implemented and tested in a mobile dating case study.

1. Introduction

Powerful and innovative mobile devices enable the possibility to adopt advanced discovery techniques in ad-hoc and peer-to-peer environments. Though increasingly effective, such devices have specific limitations, which have to be taken into account when designing systems and applications able to support mobile users.

We propose a novel discovery framework whose concrete implementation has been carried out in a mobile *dating* case study even if it is cross-applicable in several discovery scenarios. Knowledge Representation techniques and approaches have been shaped to be effectively suitable in volatile ubiquitous computing contexts. In particular, we propose a revision of abduction and contraction algorithms presented in [3] in order to allow an exploitation of them in resource-constrained contexts. Building on previous work that enhanced the discovery possibilities offered by standard code-based matching procedures with semantic-based capabilities [10], here we devise a further evolution of matchmaking algorithms allowing to by-pass the presence of fixed and networked servers. This framework and approach has been applied and tested for profile matchmaking in a mobile p2p environment.

Users equipped with a mobile device expose both their semantically annotated profile and preferences they would like to satisfy encountering another user. An exact match between requester preferences and offered profiles is surely the best possible result, but it is probably too rare to be realistic. It is more feasible to obtain a ranked list of available user profiles even if they do not completely fulfill the request. In the same way, when the user preferences and retrieved profiles are incompatible, it could be interesting to know what are the causes for the incongruence if user is willing to retract some constraints she originally imposed to reach a potential match.

The proposed system exploits a revised version of nonmonotonic inferences (in particular abduction and contraction) [4] to retrieve compatible profiles arranged in relevance order. A score is computed taking into account the semantic *affinity* between preferences expressed by the user and characteristics found in the available profiles.

As explained hereafter, we selected a sublanguage deriving from OWL DL[1], $\mathcal{ALN}$(D), to model ontologies, preferences and profile annotations whereas the proposed system adopts DIG 2.0[2] annotations.

The Bluetooth connectivity of handheld user's device is exploited to allow the data exchange. To this aim the basic service discovery protocol has been extended with semantic capabilities: a "micro-layer" has been integrated within a J2ME[3] application level over the Bluetooth stack. In this way, a mobile host performing a query and another one exposing its characteristics can quickly communicate.

[1] OWL Web Ontology Language, W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-features/

[2] DIG 2.0: The DIG Description Logic Interface, http://dig.cs.manchester.ac.uk/overview.html

[3] Java 2 Micro Edition: http://java.sun.com/javame/index.jsp

We adopt a simple piconet configuration without stable networked zone servers. Peers are equipped with a Bluetooth interface and they are at the same time able to address requests to other mobile clients as well as to receive and reply to external queries. Each device hosts a semantic facilitator to match user preferences with arrived profiles.

The remaining of the paper is structured as follows: in the next Section we outline motivation of the proposed approach and most relevant background of it; afterward, in Section 3 we move on to the presentation of the theoretical framework. Relevant features of the dating application we implemented are outlined in Section 4 with the aid of a simple illustrative case study. Conclusions close the paper.

2. Background and Motivation

Non-logical approaches to resource retrieval have serious limitations. First of all, exploiting standard relational database techniques to model a resource retrieval framework, there is the need to completely align the attributes of the offered and requested resources descriptions, in order to perform a match. If requests and offers are simple names or strings, the only possible match would be identity, resulting in an all-or-nothing outcome of the retrieval process. Vague query answering, proposed by [9], was an initial effort to overcome the rigid constraints of relational databases, by attributing weights to several search variables.

Vector-based techniques taken by classical Information Retrieval can be used too, thus reverting the search for a resource matching a request to similarity between weighted vectors of stemmed terms, as proposed in the COINS matchmaker [8] or in LARKS [11]. 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 in Smart [1] or hybrid approaches, as in the OWLS-MX matchmaker [7].

Further approaches structure resource descriptions as set of words. This formalization allows one to evaluate not only identity between sets, but also some interesting set-based relations between descriptions, such as inclusion, partial overlap, or cardinality of set difference. Anyway, modeling resource descriptions as set of words is too much sensitive to the choice of employed words to be successfully used: the fixed terminology misses meaning that relates to the words. Such a problem can be solved giving to terms a logical and shared meaning through an ontology [5]. Nevertheless set-based approaches already have some properties we believe are fundamental in a resource 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, because they completely fulfill the resource to retrieve. Moreover even if there are characteristics of the retrieved resource not elicited in the description of the searched one, an exact match is still possible because absent information has not to be considered negative. The two statements above may be summarized in the so called *Open World Assumption* (OWA). That is 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 requester.

DL-based systems usually provide two basic reasoning services for a TBox $\mathcal{T}$ (i.e. an ontology), namely *Satisfiability* and *Subsumption*. Both are adequate only in scenarios where a simple yes/no answer is enough. Nevertheless, often a more flexible discovery approach is needed also coping with non-exact matches. To this aim, in [2] *Concept Contraction* and *Concept Abduction* non-standard inference services for DLs were introduced and defined, resulting in an application on a wired-side fixed Description Logics reasoning engine, in a subset of OWL-DL ($\mathcal{ALN}$)[4].

Anyway, in order to adapt ideas and techniques of semantic-based retrieval approaches developed for Web contexts to ubiquitous computing, one has to take computational costs into account. Expressiveness of adopted formalisms plays a critical role. Hence w.r.t. applications which involve fixed reasoners, in mobile contexts care has to be paid in reducing the concrete impact of the inference procedures on the mobile devices. So algorithms and approaches must be modified and adapted to be suitable for applications involving computationally limited components with high volatility. In those cases it can be more appropriate to shape "agile" and fast reacting discovery architectures rather than computationally heavy structures somewhat uncontrollable.

3. Framework and Approach

The proposed approach has been devised and tested for mobile devices simply equipped with a Bluetooth connectivity. No specific requirements in terms of available memory and computational capabilities have to be satisfied. Each mobile device in the environment participates to a given Bluetooth piconet. To save memory (which is a precious resource in *handhelds*), user profile and preferences are expressed in DIG formalism –a more compact variant of OWL. When two or more devices are in the same radio range (approximately 100 mt for class I Bluetooth devices at 20 dBm power), a user can start a discovery procedure aiming to identify hosts wanting to interact.

In what follows will be analyzed the algorithms featuring the matchmaking process in the mobile scenarios described

[4] MaMaS-TNG (MatchMaking Service-The Next Generation), available as an HTTP service at: http://dee227.poliba.it:8080/MAMAS-tng/DIG

above. Each request and supply (from now on D and S) as well as the given TBox $\mathcal{T}$ will be expressed in $\mathcal{ALN}$(D) Description Logic[5].

3.1 Logic-based matchmaking via Concept Abduction and Contraction

In real scenarios, it is quite rare to determine exactly the service/resource/profile we are looking for. Often we have to reformulate the request in order to obtain satisfactory results in an approximate search. So a question arises: *what* should we change? Some suggestions would be useful.

Let $\mathcal{L} = \mathcal{ALN}(D)$, S, D be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$. Given S and D, the matchmaking procedure can happen as detailed here:

- if the conjunction $S \sqcap D$ is unsatisfiable in the TBox $\mathcal{T}$ representing the ontology, *i.e.*, they are not compatible with each other, the requester can retract some requirements in D, G (for *Give up*), to obtain a concept K (for *Keep*) such that $K \sqcap S$ is satisfiable in $\mathcal{T}$ (Concept Contraction Problem, CCP).

- if S and D are compatible with each other, it could be the case that S –though compatible– does not imply D. That is in DL syntax: $\mathcal{T} \models D \sqcap S \not\sqsubseteq \bot$ and $\mathcal{T} \models S \not\sqsubseteq D$. Then, it is necessary to assess what should be hypothesized (H) in S in order to completely satisfy D (Concept Abduction Problem, CAP).

Given an offered service/resource/profile S, both Concept Abduction and Concept Contraction can be used to suggest guidelines on what has to be revised and/or hypothesized in order to obtain a full match with the request. We now show how the previously introduced reasoning services can help in an approximate semantic-based search of resources, fully exploiting their structured description.

Let us suppose to have a request D, a resource S and an ontology $\mathcal{T}$ such that $\mathcal{T} \models D \sqcap S \not\sqsubseteq \bot$, *i.e.*, they are incompatible. In order to gain compatibility, a Concept Contraction is needed so that giving up G in D, the remaining K could be satisfied by S. Now, if $\mathcal{T} \not\models S \sqsubseteq K$, the solution H_K to the CAP $\langle \mathcal{L}, K, \mathtt{S}, \mathcal{T} \rangle$ represents what is in K and is not specified in S.

As the S obtained is an approximated match of D, then it would be extremely useful to evaluate how good the approximation is. Given more than one service/resource/profile, which is the best approximation? How can a numerical score be assigned to the approximation, based on K, H and G, in order to rank the resources? Algorithm 1 provides answers to the raised issues[6].

Algorithm: $explain(\mathtt{S}, \mathtt{D}, \mathcal{T}, \mathcal{L})$

Input: S, D concepts in $\mathcal{L}$ such that $\mathcal{T} \models S$ and $\mathcal{T} \models D$

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

```
1 if T ⊨ D ⊓ S ⊑ ⊥ then
2     ⟨G, K⟩ = contract(S, D, T);
3     H_K = abduce(S, K, T);
4     return ⟨G, K⟩, H_K;
5 else
6     H = abduce(S, D, T);
7     return ⟨⊤, D⟩, H;
8 end
```

Algorithm 1: Matching explanation

A brief explanation of the above algorithm will clarify its rationale:

[lines 1-4] Having a request D and a supplied service/resource/profile S, 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 D is performed in order to regain compatibility [row 2] and then the hypothesis on S is computed in order to completely satisfy D (or its contraction) [row 3].

[lines 5-7] If the conjunction of D's and S'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.

Notice that $H = abduce(\mathtt{S}, \mathtt{D}, \mathcal{T})$ [rows 3,6] determines a solution H for the CAP $\langle \mathcal{L}, \mathtt{D}, \mathtt{S}, \mathcal{T} \rangle$, while $\langle G, K \rangle = contract(\mathtt{S}, \mathtt{D}, \mathcal{T})$ [row 2] determines a solution $\langle G, K \rangle$ for the CCP $\langle \mathcal{L}, \mathtt{D}, \mathtt{S}, \mathcal{T} \rangle$. The algorithm $explain$ returns values useful in a retrieval system where explanation of results is needed and/or a belief revision process is admitted. It does not depend on the particular DL adopted. Based on the minimality criteria proposed in [2] the length $|H|$ of the solution to a CAP for $\mathcal{ALN}$(D) DL can be computed. Hence, a relevance ranking score can be computed.

3.2 Algorithms details

The matchmaking starts when are available the unfolded versions of D and S, expressed in Conjunctive Normal Form (CNF) [4]. The *Unfolding* and *Normalization* procedures aiming to pre-process D and S are outlined in the following subsections. Furthermore, the Abduction and Contraction algorithms are reported.

---

[5]We assume the reader be familiar with basics of DL formalisms and we refer her to [4, 3] for several examples and wider argumentation.

[6]In the algorithm and in what follows, with $\top$ we refer to the most generic concept in an ontology and 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.

3.2.1 Concept Unfolding

Given the $\mathcal{ALN}$(D) DL, an acyclic TBox $\mathcal{T}$ and a concept C in $\mathcal{ALN}$(D), the *unfolding* procedure is performed by applying the following substitutions to build a new unfolded concept equivalent to C w.r.t. $\mathcal{T}$.

$A \rightarrow A \sqcap C$ if $A \sqsubseteq C \in \mathcal{T}$
$A \rightarrow C \; if \; A \equiv C \in \mathcal{T}$
$A \rightarrow A \sqcap \neg B_1 \sqcap ... \sqcap \neg B_k \; if \; disj(A, B_1, ..., B_k) \in \mathcal{T}$

3.2.2 CNF

The *Conjunctive Normal Form* of a concept can be obtained by applying the following substitutions (after the unfolding procedure) until no more substitutions are possible:

$C \sqcap \bot \rightarrow \bot$
$(\geq n\, R) \sqcap (\leq m\, R) \rightarrow \bot$ if $n > m$
$A \sqcap \neg A \rightarrow \bot$
$(\geq n\, R) \sqcap (\geq m\, R) \rightarrow (\geq n\, R)$ if $n > m$
$(\leq n\, R) \sqcap (\leq m\, R) \rightarrow (\leq n\, R)$ if $n < m$
$\forall R.D_1 \sqcap \forall R.D_2 \rightarrow \forall R.\,(D_1 \sqcap D_2)$
$\forall R.\bot \rightarrow \forall R.\bot \sqcap (\leq 0\, R)$

3.2.3 Concept Contraction

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

We note that there is always the trivial solution $\langle G, K \rangle = \langle D, \top \rangle$ to a CCP. This solution corresponds to the most drastic contraction, that gives up everything of D. In our resource retrieval framework, it models the (infrequent) situation where, in front of some very appealing profile S, incompatible with the requested one, a user just gives up completely his/her specifications D in order to meet S.

On the other hand, when $S \sqcap D$ is satisfiable in $\mathcal{T}$, the "best" possible solution is $\langle \top, D \rangle$, 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 criteria in the contraction must be defined [6]. 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 to be taken into account. For example, we can think to relax some constraints in the original D introducing a *penalty* given by the following function $\prod_C = f(p_D, p_S)$ where p_D e p_S are the conflicting predicates respectively within the demand and the supply.

Algorithm: $Contract\,(\langle \mathcal{L}, D, S, \mathcal{T} \rangle)$
Require: $\langle \mathcal{L}, D, S, \mathcal{T} \rangle$ with $\mathcal{L} = \mathcal{ALN}(D)$, acyclic $\mathcal{T}$
Ensure: $\langle G, K \rangle$ with concept G, K in $\mathcal{L}$

- **if** $D == \bot$ **then**
  - **return** $\langle \bot, \top \rangle$
- **else**
  - $G := \top$
  - $K := \top \sqcap D$
  - **for** each concept name A in K **do**
    - $U := unfolding\,(\langle \mathcal{L}, A, \mathcal{T} \rangle)$
    - **for** each concept name A' in U **do**
      - **if** exists B in S such that $B = \neg A'$ **then**
        - $G := G \sqcap A$
        - remove A from K
      - **end if**
    - **end for**
  - **end for**
  - **for** each concept $C := (\geq\ x\ R)$ in K **do**
    - **if** exists $(\leq\ y\ R)$ in S and $y < x$ **then**
      - replace C with $(\geq\ y\ R)$
      - $G := G \sqcap C$
    - **end if**
    - **for** each concept $\forall R.E$ in K **do**
      - **if** exists $\forall R.F$ in S **then**
        - $\langle G',\, K' \rangle := Contract\,(\langle \mathcal{L}, E, F, \mathcal{T} \rangle)$
        - $G := G \sqcap \forall R.G'$
        - replace $\forall R.E$ in K with $\forall R.K'$
      - **end if**
    - **end for**
  - **end for**
  - **for** each concept $C := (\leq\ x\ R)$ in K **do**
    - **if** exists $(\geq\ y\ R)$ in S and $y > x$ **then**
      - replace C with $(\leq\ y\ R)$
      - $G := G \sqcap C$
    - **end if**
  - **end for**
  - **for** each predicate $\geq_x g$ in K **do**
    - **if** exists $\leq_y g$ in S and $y < x$ **then**
      - replace $\geq_x g$ with $\geq_y g$
      - $G := G \sqcap C$
    - **end if**
  - **end for**
  - **for** each predicate $\leq_x g$ in K **do**
    - **if** exists $\geq_y g$ in S and $y > x$ **then**
      - replace $\leq_x g$ with $\leq_y g$
      - $G := G \sqcap C$
    - **end if**
  - **end for**
- **end if**
- **return** $\langle G,\, K \rangle$

3.2.4 Concept Abduction

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

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

Also in this case, one can think to a *penalty function* which increases when the number of missing concepts in S w.r.t. to D grows. The penalty function for the abduction problem can be expressed as $\prod_A = f\,(p_D, p_S)$ where p_D e p_S are missing predicates or restrictions respectively in D and S.

Algorithm: *Abduce* $(\langle \mathcal{L}, D, S, \mathcal{T} \rangle)$
Require: $\langle \mathcal{L}, D, S, \mathcal{T} \rangle$ with $\mathcal{L} = \mathcal{ALN}(D)$, acyclic $\mathcal{T}$
Ensure: H in $\mathcal{L}$

$H := \top$
for each concept name A in D **do**
  if does not exist B in S such that $B \sqsubseteq A$ **then**
    $H := H \sqcap A$
  end if
end for
for each concept $(\geq\ x\ R)$ in D **do**
  if does not exist $(\geq\ y\ R)$ in S with $y \geq x$ **then**
    $H := H \sqcap (\geq\ x\ R)$
  end if
end for
for each concept $(\leq\ x\ R)$ in D **do**
  if does not exist $(\leq\ y\ R)$ in S with $x \geq y$ **then**
    $H := H \sqcap (\leq\ x\ R)$
  end if
end for
for each concept $\geq_x\ g$ in D **do**
  if does not exist $\geq_y\ g$ in S with $y \geq x$ **then**
    $H := H \sqcap \geq_x\ g$
  end if
end for
for each concept $\leq_x\ g$ in D **do**
  if does not exist $\leq_y\ g$ in S with $x \leq y$ **then**
    $H := H \sqcap \leq_x\ g$
  end if
end for
for each concept $\forall R.E$ in D **do**
  if exists $\forall R.F$ in S **then**
    $H' :=$ *Abduce* $(\langle \mathcal{L}, E, F, \mathcal{T} \rangle)$
    $H := H \sqcap \forall R.H'$
  else
    $H := H \sqcap \forall R.E$
  end if
end for
return H

4. Illustrative example

A mobile dating application was developed as a case study for the proposed mobile matchmaking framework and algorithms. The goal of the application is to facilitate acquaintance among people in a given environment. It is compatible with the Mobile Information Device Profile (MIDP[7]) of Java ME (Micro Edition) platform and uses Bluetooth connectivity. Both features are currently granted by the wide majority of mobile phones. No external infrastructure support is needed for discovery, matchmaking and interaction among users: the proposed application is a true peer-to-peer ubiquitous computing application, based only on Bluetooth wireless ad-hoc networking and a on built-in reasoning engine. Each user stores a personal *profile* and a *demand* for desired partner characteristics on his/her device[8]. Both are defined as $\mathcal{ALN}$(D) conjunctive concept expressions referring to a common domain ontology, which models people's physical appearance and personal interests. Due to lack of space, axioms in the reference ontology are not reported here.

The main features of the application we implemented are summarized in Figure 1: **(I)** peers in communication range are discovered; **(II)** user profiles are exchanged and match degree is computed (implementing the algorithms described above) between each user's demand and the peer's profile; **(III)** if both peers are satisfied with the match outcome, they start a chat session.

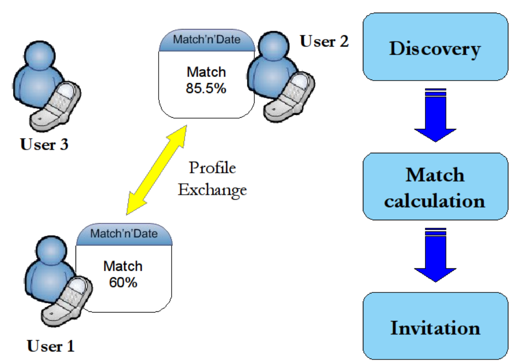

Figure 1. Feature summary of the proposed application

In what follows a case study is presented to make clear the proposed approach along with its implementation.

Albert has been invited to a party by his friend Joe, but he is getting quite bored. Joe is spending all the time with his

[7] http://java.sun.com/products/midp/

[8] On-device editing of profile and demand is not supported at this moment, it is planned as an upcoming application feature. User profiles are currently defined through an external tool and then transferred to the mobile device.

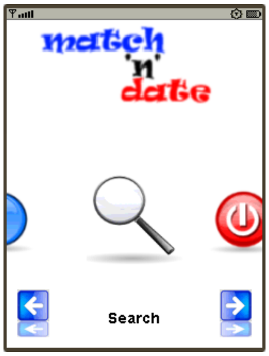

Figure 2. Main application form

Figure 3. Matchmaking score form

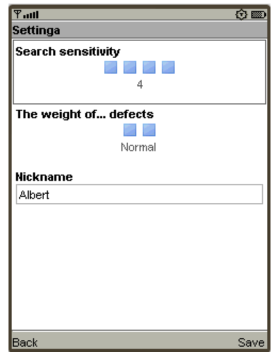

Figure 4. Settings form

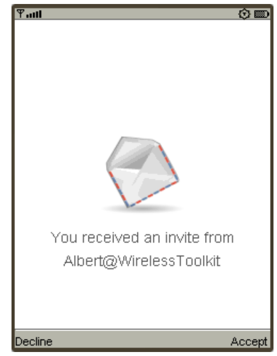

Figure 5. Invite notification form

girlfriend and Albert does not know anyone and he cannot find interesting conversation topics with other people. He would like to find a nice girl to talk to, so he launches the application on his mobile phone. It loads user profile and demand, then main menu is shown as in Figure 2. User start selects "discovery" and the application uses standard Bluetooth SDP (Service Discovery Protocol) to search for other instances of it running on remote mobile devices in Bluetooth radio range.

A remote device running the same application is found. It belongs to Barbara, which is getting bored too. Albert's device retrieves Barbara's user profile in order to perform matchmaking with characteristics in Albert's demand preference. Profile retrieval is performed by means of standard Bluetooth OBEX (OBject EXchange) protocol[9]. Matchmaking is executed by an integrated mobile reasoning engine upon $\mathcal{ALN}$(D) simple-TBoxes, as explained in Section 3. The matchmaker was implemented as a Java ME class package.

Let us suppose that Albert is looking for a woman between 21 and 32 years old and between 160 and 180 cm high, which likes walking and –more important– has a passion for swimming. A crucial requirement is that she is not already engaged. These preferences can be expressed formally in the semantic-based matchmaking framework described in Section 3, referring to the dating domain ontology:

D_{Albert}: $\geq_{age} 21 \sqcap \leq_{age} 32 \sqcap \geq_{height} 160 \sqcap \leq_{height} 180 \sqcap \exists hasMaritalState \sqcap \forall hasMaritalState.(free \sqcap =_{relevance} 1) \sqcap \exists hasHobby \sqcap \forall hasHobby.(walking \sqcap =_{relevance} 5) \sqcap \exists hasSportPassion \sqcap \forall hasSportPassion.(swimming \sqcap =_{relevance} 3)$

Barbara's profile can be modeled likewise. She is 28 years old and 172 cm high. She has red hair and is has free marital status. She likes motorcycling but not swimming, pop-rock music and romantic movies but not science fiction ones. Formally, this is expressed as:

$P_{Barbara}$: $=_{age} 28 \sqcap =_{height} 172 \sqcap \exists hasHairColor \sqcap \forall hasHairColor.red \sqcap \exists hasMaritalStatus \sqcap \forall hasMaritalStatus.free \sqcap \exists hasSportPassion \sqcap \forall hasSportPassion.(motorcycling \sqcap \neg swimming) \sqcap \exists favoriteMusicGenre \sqcap \forall favoriteMusicGenre.pop-rock \sqcap \exists favoriteMovieGenre \sqcap \forall favoriteMovieGenre.(romantic \sqcap \neg sci-fi)$

As depicted in Figure 3, after matchmaking the user is presented with an overall match score s, ranging from 0 to 100. It is computed by means of the formula:

$$s = 100\% \begin{cases} 1-p & 1-p > T \\ 0 & otherwise \end{cases}$$

where the semantic *penalty function* p is computed as:

$$p = W * contract + (1-W) * abduce$$

where $contract$ is the penalty calculated by the contraction procedure between the local user's demand D and the remote user's profile P, while $abduce$ is the penalty value of the abduction procedure between the consistent part K of the demand and P. The scoring mechanism is regulated by two parameters, threshold value T and weight W, both ranging from 0 to 1. T influences the sensitivity of the discovery. W determines the relative weight of explicitly conflicting elements in P with respect to D, which can be seen as "defects" from the requester's standpoint. Both parameters can be modified in the *Settings* screen shown in Figure 4, along with the nickname that appears to other users during profile exchange, invitation and chat.

In our example, with $W = 0.315$ and $T = 0.6$, outcomes are as follows:

G (Give up): $\forall hasSportPassion.(swimming \sqcap =_{relevance} 3)$

[9]Further details about Bluetooth interaction protocol have been omitted here due to lack of space.

K (Keep): $\geq_{age} 21 \sqcap \leq_{age} 32 \sqcap \geq_{height} 160 \sqcap \leq_{height} 180 \sqcap \exists hasMaritalState \sqcap \forall hasMaritalState.(free \sqcap =_{relevance} 1) \sqcap \exists hasSportPassion \sqcap \exists hasHobby \sqcap \forall hasHobby.(walking \sqcap =_{relevance} 5)$
H (Hypothesis): $\sqcap \exists hasHobby \sqcap \forall hasHobby.(walking \sqcap =_{relevance} 5)$
$contract = 0.0625$
$abduce = 0.4$
$p = 0.2938$
$s = 70.62\%$
Notice that the restriction on sport to swimming is explicitly inconsistent with the supplied profile, so it must be given up through concept contraction, while the remaining part of the demand can be kept. Of such K, the restriction to walking as a hobby is not stated in the supply, so concept abduction computes it as a hypothesis.

Albert is satisfied with the match outcome and wishes to invite Barbara to a chat. The dating application allows the user to contact the remote device for a chat session.

A simple text-based protocol was developed on top of Bluetooth OBEX for this purpose. Upon reception of an invite from Device A, Device B displays a notification to the user (like the one reported in Figure 5), who can either decline immediately or let her device retrieve the peer's profile from Device A and compute the match degree with her own demand specification. The result is shown to the user of Device B and then she decides whether to accept or decline the invitation. If she accepts, the chat session starts.

Barbara receives the invite from Albert. Her application instance retrieves his profile and shows to her the match score with demanded characteristics. She views the result and accepts the invite, so the chat session can start. Chat helps them make eye-contact amidst the party crowd. Hopefully, this is the beginning of a new friendship.

5. Conclusion

We have proposed a novel discovery framework for mobile ad-hoc contexts without stable and fixed network infrastructures. Abduction and contraction algorithms presented in [4] have been adapted to allow an exploitation in wireless and p2p scenarios. The proposed approach has been validated in a dating case study where users – equipped with a Bluetooth device– search for semantically annotated profiles in the environment compatible with their preferences (also expressed by means of a logic annotation). Framework and approach are general purpose as they are fully re-usable in different contexts and applications.

Future work is aimed at enhancing the expressiveness of the managed logic attempting to remove some constraint actually imposed. We are currently working on a thorough evaluation of the approach basically measuring the response times of the system in different use cases and with different hardware and network configurations.

Acknowledgments

The authors acknowledge partial support of Apulia Region Strategic Projects PS_092 and PS_121.

References

[1] S. Agarwal and S. Lamparter. smart - a semantic matchmaking portal for electronic markets. In *Proceedings of the 7th International IEEE Conference on E-Commerce Technology 2005*, 2005.

[2] 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. *Electronic Commerce Research and Applications*, 4(4):345–361, 2005.

[3] 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*, 12(2):127–154, 2007.

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

[5] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

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

[7] M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with owls-mx. In *In AAMAS 2006*, pages 915–922. ACM Press, 2006.

[8] D. Kuokka and L. Harada. Integrating Information Via Matchmaking. 6:261–279, 1996.

[9] A. Motro. VAGUE: A User Interface to Relational Databases that Permits Vague Queries. *ACM Trans. Office Inf. Syst.*, 6(3):187–214, 1988.

[10] M. Ruta, T. Di Noia, E. Di Sciascio, and F. Donini. Semantic based collaborative p2p in ubiquitous computing. *Web Intelligence and Agent Systems*, 5(4):375–391, 2007.

[11] 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.