Semantic Based Collaborative P2P in Ubiquitous Computing

Michele Ruta [1] Tommaso Di Noia [2] Eugenio Di Sciascio [3]

Politecnico di Bari
via Re David 200, I-70125 Bari, Italy

Francesco Maria Donini [4]

Università della Tuscia
via San Carlo 32, I-01100 Viterbo, Italy

Abstract

We present a collaborative environment for semantic-enabled mobile devices (*e.g.*, PDAs, cell phones, laptops) in peer to peer scenarios. Within the environment, resource discovery is performed exploiting technologies and techniques for knowledge representation developed for the Semantic Web, which have been adapted to cope with the highly flexible structure of ad-hoc networks in ubiquitous computing. The approach exploits the standard Bluetooth stack, using the original UUID payload, to carry semantically annotated data. The framework is motivated and presented in a museum case study.

1 Introduction

The growth in the diffusion of wireless-enabled handheld devices provides the necessary infrastructure for creating ad-hoc environments for ubiquitous computing. In such a mobile infrastructure there is one or more devices providing resources or using services.

As an ad-hoc network is a very unpredictable environment, Service Discovery (SD) becomes an essential feature. In fact in an ubiquitous context,

[1] Email: `m.ruta@poliba.it`
[2] Email: `t.dinoia@poliba.it`
[3] Email: `disciascio@poliba.it`
[4] Email: `donini@unitus.it`

information about services are often unavailable, because the location of mobile devices could change continuously [10].

A flexible service search system is desirable, based on wireless network infrastructure and able to overcome difficulties due to the host mobility.

In this paper we present a collaborative, ontology-based environment, where a semantic enabled resource discovery mechanism is employed to obtain advanced discovery features, providing automated services to users participating to the ad-hoc network.

In a semantic-enabled Web –what is known as the Semantic Web vision– each available resource should be annotated using RDF [21] with respect to an OWL ontology [15]. There is a close relation between the OWL-DL subset of OWL and Description Logics (DLs) [2] semantics, which allows the use of DLs based reasoners in order to infer new information from the one available in the annotation itself. In this paper we will refer to DIG [5] instead of OWL-DL because it is less verbose and more compact: a good characteristic in an ad-hoc scenario. DIG can be seen as a syntactic variant of OWL-DL.

Here we use some results and approaches borrowed from Semantic Web studies, to extend the Bluetooth Service Discovery Protocol (SDP) in order to provide semantic-based services to the users within the piconet.

The original Bluetooth standard uses SDP at the application layer. It is code-based basically, hence it can handle only exact matches. Yet, if we want to retrieve resources whose description cannot be classified within a rigid schema (*e.g.*, the description of paintings in a museum or goods in a shopping mall) a more flexible SDP is needed. Such a SDP must be able also to cope with non-exact matches, to provide a ranked list of discovered resources computing a distance between each retrieved resource and the requested one, after a matchmaking process. To achieve these goals, we exploit both theoretical approach and technologies of Semantic Web vision and adapt them to small ad-hoc networks based on the Bluetooth technology [22].

Actually, a set-value based approach could satisfy, at a first glance, similar SDP requirements, but imagine the following simple example, related to a museum. Suppose you are looking for paintings whose subject is a portrait, $R = \{portrait\}$, and in the museum there is a collection of self-portraits as offered resources, $O = \{selfportrait\}$, the two values do not match and nothing is known about their relations. Hence, no resource will be retrieved to answer your request. To overcome similar, and more complex, problems we need to model the meaning of the terms and their relations. That is, a representation of their semantics is needed. We believe the SDP efficiency and flexibility can be enhanced by exploiting knowledge representation techniques.

The rest of the paper is structured as follows: next section introduces basics of SDP, DLs and OWL/DIG; in Section 3 we present the framework of our approach, whereas in Section 4 the semantic-enabled SDP is outlined;

in Section 5 we explain system features and behavior by means of a case study deployed in a museum; in Section 6 further P2P collaborative semantic services are sketched. In Section 7 we comment on related works and finally we outline conclusion and future work in Section 8.

2 Basics

2.1 SDP

Usually, resource discovery protocols involve a requester, a lookup or directory server and finally a resource provider. Most common SDPs, Service Location Protocol (SLP), Jini, UPnP (Universal Plug aNd Play), Salutation or UDDI (Universal Description Discovery and Integration) among others, include registration and lookup of resources as well as matching mechanisms [4].

All these systems generally work in a similar manner. Basically a client issues a query to a directory server or to a specific resource provider. The request may explicitly contain a resource name with one or more attributes. The lookup server –or directly the resource provider– attempts to match the query pattern with resource descriptions stored in its database, then it replies to the client with discovered resources identification and location [18].

These discovery architectures are based on some common assumptions about network infrastructure under the application layer in the protocol stack. In particular, current SDPs usually require a continuous and robust network connectivity, which may not be the case in wireless contexts, especially in the ad-hoc ones. In fact, in such environments network consistence varies continuously and temporary disconnections occur frequently, bringing to a substantial decrease to traditional SDP performances [10].

Actually there are several issues that restrain the expansion of advanced wireless applications. Among them, the variability of scenarios. Basically an ad-hoc environment is based on technologies able to grant the peer to peer interaction among variously located clients. In such a mobile infrastructure there could be one or more devices providing and using resources or services but, as a MANET is a very unpredictable environment, a flexible resource search system is needed to overcome difficulties due to the host mobility. Furthermore, existing mobile resource discovery methods use a simple string-matching, which is largely inefficient in advanced scenarios where there is the need to submit articulate requests to the system, to obtain adequate responses [23].

With specific reference to the SDP in the Bluetooth stack, it is based on a 128 bit Universally Unique Identifier (UUID); each numeric ID is associated to a single service class. Resource matching in Bluetooth is hence strictly syntactic and consequently SDP manages only exact matches.

2.2 DLs, OWL and DIG

In this subsection we summarize notions and definitions about Description Logics, OWL and DIG we exploit in our approach.

Description Logics (DLs) are a family of logic formalisms for Knowledge Representation [7,14], also knows as Terminological languages, as Concept languages, in a subset of First Order Logic. In DLs, the basic syntax elements are:

- *concept* names, *e.g.*, `Painting`, `Sculpture`, `Style`
- *role* names, like `hasStyle`
- *individuals*, like *Broyeuse of Chocolat*

Intuitively, concepts stand for sets of objects, and roles link objects in different concepts, *e.g.*, the role `hasStyle` links paintings to styles. Individuals are used for special named elements belonging to concepts.

A semantic *interpretation* is a pair $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, consisting 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. We assume that different individuals are mapped to different elements of Δ, *i.e.*, if $a \neq b$ then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. This restriction is usually called *Unique Name Assumption* (UNA).

Previous basic elements can be combined using *constructors* to form concept and role *expressions*. Each DL has a different set of constructors. A constructor used in every DL is the one allowing the *conjunction* of concepts, usually denoted as $\sqcap$; some DL include also disjunction $\sqcup$ and complement $\neg$ to close concept expressions under boolean operations.

Roles can be combined with concepts using *existential role quantification* (*e.g.*, $\texttt{Bust} \sqcap \exists \texttt{madeOf.Bronze}$, which indicates the set of busts whose materials include bronze) and *universal role quantification* (*e.g.*, $\texttt{Painting} \sqcap \forall \texttt{hasStyle.Dadaism}$, which describes only dadaist paintings). Other constructs may involve counting, as *number restrictions*: $\texttt{Fresco} \sqcap (\leq 1\, \texttt{hasAuthor})$ expresses frescos with just one author, and $\texttt{Fresco} \sqcap (\geq 2\, \texttt{hasAuthor})$ describes frescos created by at least two artists.

Many other constructs can be defined, up to create n-ary relations [9], so increasing the expressiveness of the DL.

Semantics of the expressions is given defining the interpretation function over each construct. For example, concept conjunction is interpreted as set intersection: $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, whereas the other connectives $\sqcup$ and $\neg$, if present, maintain the usual theoretical interpretation of $\cup$ operator and complement one. The interpretation of constructs involving role quantification needs to make explicit domain elements: $(\forall R.C)^{\mathcal{I}} = \{d_1 \in \Delta \mid \forall d_2 \in \Delta : (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$

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. For example, we could impose that portraits can be divided into selfportraits and likenesses using the two inclusions: $\texttt{Portrait} \sqsubseteq \texttt{SelfPortrait} \sqcup \texttt{Likeness}$ and $\texttt{SelfPortrait} \sqsubseteq \neg\texttt{Likeness}$, or that oil paintings are made using only one pictorial technique as $\texttt{OilPainting} \sqsubseteq (\leq\ 1\ \texttt{hasTechnique})$.

Definitions are useful to give a meaningful name to particular combinations, as in $\texttt{DadaistPainting} \equiv \texttt{Painting} \sqcap \forall\texttt{hasStyle.Dadaism}$. Sets of such inclusions are called TBox (Terminological Box). In simple DLs, only a concept name can appear on the left-hand side of an inclusion.

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 T is an interpretation satisfying all inclusions and definitions of T.

Adding new constructors makes DL languages more expressive. Nevertheless, this usually leads to a growth in computational complexity of inference services [8]. Hence a trade-off is worthwhile.

The basic idea of the Semantic Web initiative is to annotate information by means of markup languages, based on XML, such as RDF and RDFS [21], DAML+OIL [12,19] and more recently OWL [15]. These languages have been conceived to allow machine understandable, unambiguous representation of web contents through the creation of domain ontologies, increasing openness and interoperability in the WWW. The strong relations between DLs and the above referenced languages for the Semantic Web [3] is also evident in the definition of the OWL language. In fact there are three different sub-languages for OWL:

- *OWL-Lite*. It allows class hierarchy and simple constraints on relation between classes.
- *OWL-DL*. Based on DLs theoretical studies, it allows a great expressiveness keeping computational completeness and decidability.
- *OWL-Full*. Using such a language, there is a huge syntactic flexibility and expressiveness. This freedom is paid in terms of no computational guarantee.

In this paper we will refer to the *Attributive Language with unqualified Number restrictions* ($\mathcal{ALN}$) DL, a subset of OWL-DL [15]. Constructs of $\mathcal{ALN}$ DL are reported in what follows (see Table 1 for further details):

- $\top$, *universal concept*. All the objects in the domain.
- $\bot$, *bottom concept*. The empty set.
- A, *atomic concepts*. All the objects belonging to the set A.

- $\neg A$, *atomic negation*. All the objects not belonging to the set A.
- $C \sqcap D$, *intersection*. The objects belonging both to C and D.
- $\forall R.C$, *universal restriction*. All the objects participating in the R relation whose range are all the objects belonging to C.
- $\exists R$, *unqualified existential restriction*. There exists at least one object participating in the relation R.
- $(\geq \ n \ R)$[5], $(\leq \ n \ R)$, $(= \ n \ R)$[6], *unqualified number restrictions*. Respectively the minimum, the maximum and the exact number of objects participating in the relation R.

name	syntax	semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
atomic negation	$\neg A$	$\Delta^{\mathcal{I}} \backslash A^{\mathcal{I}}$
universal quantification	$\forall R.C$	$\{d_1 \mid \forall d_2 : (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$
number restrictions	$(\geq \ n \ R)$	$\{d_1 \mid \sharp\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\} \geq n\}$
	$(\leq \ n \ R)$	$\{d_1 \mid \sharp\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\} \leq n\}$

Table 1
Syntax and semantics of $\mathcal{ALN}$constructs

Ontologies are usually designed as *simple-TBox* in order to express the relations among objects in the domain. With a *simple-TBox* the left side is represented by a concept name in all the axioms (for both inclusion and definition). See Table 2 for details.

(i) definition `OilPainting` $\equiv$ `Painting` $\sqcap \forall$`hasTechnique.Oil`
(ii) inclusion `WorkOfArt` $\sqsubseteq (\geq \ 1$ `hasAuthor`$)$

name	syntax	semantics
definition	$A = C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
inclusion	$A \sqsubseteq C$	$A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$

Table 2
Syntax and semantics of the TBox assertions

As part of the activity of the Description Logic Implementation Group (DIG) a new interface for DL systems has been defined. The DIG interface

[5] Notice that $\exists R$ is equivalent to $(\geq \ 1 \ R)$
[6] We write $(= \ n \ R)$ for $(\geq \ n \ R) \sqcap (\leq \ n \ R)$

uses HTTP as the underlying transfer protocol. It allows client (and server) developers to use existing libraries for implementation.

For DIG requests, the protocol imposes to adopt HTTP POST. The body of the request must be an XML encoded message corresponding to a DIG request.

The original DIG specification concept language is based on $\mathcal{SHOIQ}(\mathcal{D})$, that is a description logic that includes boolean concept operators (and, or, not), universal and existential restrictions, cardinality constraints, a role hierarchy, inverse roles, the one-of construct and concrete domains. For our purposes, we adopted the $\mathcal{ALN}$ description logic, which has a polynomial complexity, both for standard and non-standard inferences.

There is a strict correspondence among OWL, DIG and DL syntax as shown in the following Table 3. Nevertheless in the implementation of the proposed system we will use only DIG formalism in expressing demands as well as resource descriptions, because it is less verbose and more compact, a mandatory requirement in mobile ad-hoc applications.

In the rest of the paper we will formalize examples by adopting DL syntax instead of OWL-DL or DIG ones for compactness. Nevertheless all the semantically annotated resources as well as the ontology employed to model them, can be easily rewritten using OWL-DL or DIG formalisms.

3 Framework and Approach

In [1] it was observed that Bluetooth SDP is largely inefficient when it comes to complex requests. This is a restriction in view of the transmission capabilities increase, devised in new drafts of the standard. A more advanced service discovery protocol is desirable, able to cope with semantic descriptions associated to resources rather than simple numeric identifiers. In the framework we present, after a wireless client has been identified within the piconet, it is able to share and retrieve information from other hosts. In a typical configuration, a user contacts the zone service provider (*hotspot*) and submits a request about her interests. The server identifies clients able to share services and replies with found services, possibly ranked in a list according to their degree of correspondence to the demand.

The zone server classifies services contents by means of an ontology and users submit semantically described requests. Hence the hotspot collects the descriptions of the available resources (modeled using DLs) and computes the matchmaking rank between the request and available resources. The provided result is a ranked list of offered resources potentially matching the user request.

It should be noticed that DL-based systems usually only provide two basic reasoning services:
Concept Satisfiability: given a ontology $\mathcal{T}$ and a concept C, does there exist

OWL syntax	DIG syntax	DL syntax
$< owl : Thing/ >$	$< top/ >$	T
$< owl : Classrdf : ID = "C"/ >$	$< catom\ name = "C"/ >$	C
$< owl : ObjectPropertyrdf : ID = "R"/ >$	$< ratom\ name = "R"/ >$	R
$< rdfs : subClassOf/ >$	$< impliesc >$ $< catom\ name = "C"/ >$ E $< /impliesc >$	$C \sqsubseteq E$
$< owl : equivalentClass/ >$	$< equalc >$ $< catom\ name = "C"/ >$ E $< /equalc >$	$C \equiv E$
$< owl : disjointWith/ >$	$< disjoint >$ $< catom\ name = "C1"/ >$ $< catom\ name = "C2"/ >$ $< /disjoint >$	$C1 \neg C2$
$< owl : intersectionOf/ >$	$< and >$ $C1$ $C2$ $< /and >$	$C1 \sqcap C2$
$< owl : allValuesFrom/ >$	$< all >$ $< ratom\ name = "R"/ >$ E $< /all >$	$\forall R.E$
$< owl : maxCardinality/ >$	$< atmost\ num = "n" >$ $< ratom\ name = "R"/ >$ $< top/ >$ $< /atmost >$	$\leq\ nR$
$< owl : minCardinality/ >$	$< atleast\ num = "n" >$ $< ratom\ name = "R"/ >$ $< top/ >$ $< /atleast >$	$\geq\ nR$
$< owl : cardinality/ >$	$< and >$ $< atleast\ num = "n" >$ $< ratom\ name = "R"/ >$ $< top/ >$ $< /atleast >$ $< atmost\ num = "n" >$ $< ratom\ name = "R"/ >$ $< top/ >$ $< /atmost >$ $< /and >$	$= nR$

Table 3
Correspondence between OWL, DL and DIG syntax

at least one model of $\mathcal{T}$ assigning a non-empty extension to C?
Subsumption: given a ontology $\mathcal{T}$ and two concepts C and D, is C more general than D in any model of $\mathcal{T}$?

In a semantically-enabled resource retrieval scenario, where a matchmaking process between a request D and each of the available resources C is needed, using subsumption it is possible to establish if C is more specific than the request D, $C \sqsubseteq D$. If the previous relation holds, then the retrieved resource completely satisfies the request, *i.e.*, an *exact match* occurs. With Concept Satisfiability the discovery of incompatible resources with respect to the request can be performed. If $D \sqcap C$ is not satisfiable w.r.t. the ontology $\mathcal{T}$, the C is not compatible with the request.

Although subsumption and concept satisfiability are very useful in several scenarios for resource discovery, *exact matches* cannot be deemed the only useful, as they will be probably rare, and the service discovery results inadequate. Typically, both $D \sqcap C$ is satisfiable and $C \not\sqsubseteq D$, that is C is compatible with D but it does not completely satisfy it.

Then there is the need to go beyond subsumption and concept satisfiability to manage these frequent situations. A metric is needed to establish "how much" the resource C is compatible with the request D or, equivalently, "how much" it is not specified in C in order to completely satisfy D, that is to make the subsumption relation $C \sqsubseteq D$ true.

In [13] the *rankPotential* algorithm was proposed, such that, given a set of $\mathcal{ALN}$ axioms, $\mathcal{T}$ and two $\mathcal{ALN}$ concepts C and D both satisfiable in $\mathcal{T}$, it computes a *semantic distance* of D from C with respect to the ontology $\mathcal{T}$. Notice that we write *the distance of* C *from* D rather then *the distance between* D *and* C because of the non-symmetric behavior of *rankPotential* (see [13] for further details). In fact the relation we need to reach here is $C \sqsubseteq D$ rather then $C \equiv D$.

With the aid of *rankPotential* it is also possible to compute a complex concept depth with respect to the taxonomy represented by the axioms set $\mathcal{T}$. In fact, if $D \equiv \top$ then *rankPotential(D,C)* = *rankPotential(*$\top$*,C)* represents the distance of C from $\top$, *i.e.*, the most generic concept in the ontology. Notice that such distance is not trivially the depth of a node in a tree for at least two main reasons:

1. An $\mathcal{ALN}$ ontology, typically, is not a simple terms taxonomy tree, *i.e.*, it does not contain only IS-A relations between two atomic concepts, and can be better represented as a labeled oriented graph.

2. An $\mathcal{ALN}$ complex concept is the conjunction of atomic concepts and role expressions.

The value returned by *rankPotential(*$\top$*,C)* here represents how specific is a *complex concept expression* C with respect to an ontology $\mathcal{T}$.

Since the proposed approach is fully-compliant with Semantic Web technologies, the user exploits the same semantic enabled descriptions she may use in other Semantic Web compliant systems. That is, there is no need for different customized resource descriptions and modeling if the user uses different applications either on the web, or in mobile systems. The syntax and formal semantics of the descriptions is unique with respect to the reference ontology and can be shared among different systems.

In the next Section we outline the proposed approach to semantic discovery in Bluetooth implemented in the original SDP. Our proposal allows to reuse UUID function within Bluetooth, without troubling the original standard, and furthermore it implements an advanced P2P exchange information mechanism, where users are peer clients in the piconet, which can be both resource requesters and possible service suppliers.

4 Enhancing the Service Discovery Protocol

4.1 Infrastructure

In our mobile environment, a user contacts via Bluetooth a zone resource provider and submits her semantically annotated request in DIG formalism. For the sake of simplicity we now assume the zone server –which classifies resource contents by means of an OWL ontology– has previously identified generic hosts (wireless as well as wired) willing to promote their resources and it has already collected semantically annotated descriptions of them. Each resource in the environment owns an URI and is annotated by its OWL description.

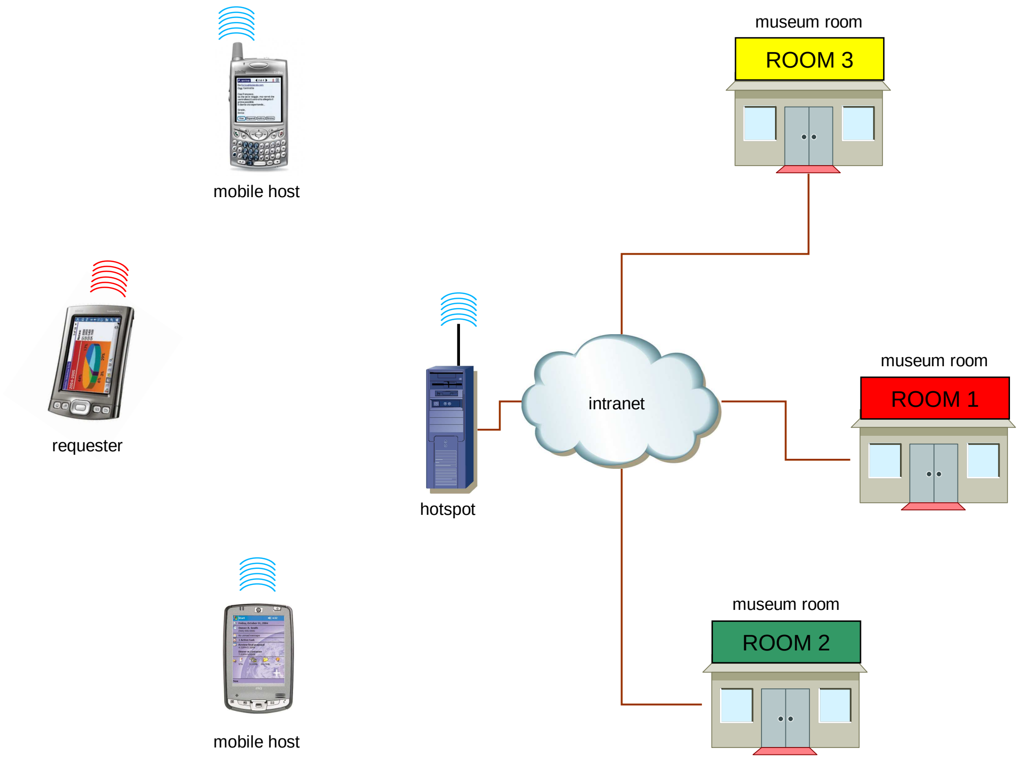

Fig. 1. A simple sketch of the infrastructure

The hotspot is endowed with a MatchMaker (in our system we adapt the *MAMAS-tng* reasoner [13]), which carries out the matchmaking process between each compatible offered resource and the requested one measuring a "semantic distance". The provided result is a list of discovered resources matching the user request, ranked according to their degree of correspondence to the request itself.

4.2 SDP modifications

The proposed framework allows the management of both syntactic and semantic discovery of resources, by integrating a semantic layer within the OSI Bluetooth stack at service discovery level. Hence, the Bluetooth standard is enriched by new functionalities which allow to maintain a backward compatibility (handheld device connectivity), but also to add the support to matchmaking of semantically annotated resources. To implement matchmaking and ontology support features, we have introduced a Semantic Service Discovery functionality into the stack, enhancing existing Bluetooth discovery protocol.

SDP uses a simple request/response method for data exchange between SDP client and SDP server [16]. We associated unused classes of 128 bit UUIDs in the original Bluetooth standard to mark each specific ontology and we call this identifier *OUUID* (Ontology Universally Unique IDentifier). In such a way, we can perform a preliminary selection of resource descriptions that do not refer to the same ontology of the request [10]. With OUUID matching we do not identify a single service, but straightforwardly the context of resources we are looking for, which can be seen as a service class.

Each semantically annotated resource is stored within the hotspot as a resource record and a 32-bit identifier is uniquely associated to each semantic resource record; we call it *SemanticResourceRecordHandle*.

OUUID	ResourceName	ResourceDescription	ResourceUtilityAttr1	ResourceUtilityAttr2	...	ResourceAttrN

Fig. 2. Scheme of resource records for semantically annotated resources

Each record contains general information about a single semantic enabled resource and it entirely consists of a list of resource attributes, see Figure2. In addition to the OUUID attribute, there are *ResourceName*, *ResourceDescription*, and a variable number of *ResourceUtilityAttr_i* attributes. *ResourceName* attribute is a text string containing a human-readable name for the resource, the second one is a text string including the resource description expressed in DIG formalism and the last ones are numeric values used according to specific applications; in general, they are associated to context-aware attributes of a resource [17]; in the current implementation we adopt, for example, the physical distance the resource has from the hotspot (expressed in meters or in terms of needed time to reach it). *ResourceUtilityAttr_i* reused as parameters of the overall utility function used to rank discovery results (see what follows for further details).

To allow the representation and the identification of a semantic annotation for a resource we introduced, within the data representation of the original

Bluetooth, two new data element type descriptors [6]. List of types is shown in Table4.

TYPE DESCRIPTOR VALUE	VALID SIZE DESCRIPTOR VALUES	TYPE DESCRIPTION
0	0	Nil, the null type
1	0, 1, 2, 3, 4	Unsigned integer
2	0, 1, 2, 3, 4	Signed twos-complement integer
3	1, 2, 4	UUID, a universally unique identifier
4	5, 6, 7	Text string
5	0	Boolean
6	5, 6, 7	Data element sequence, a data element whose data field is a sequence of data elements
7	5, 6, 7	Data element alternative, data element whose data field is a sequence of data elements from which one data element is to be selected
8	5, 6, 7	URL, a uniform resource locator
9	**1, 2, 4**	**OUUID, an ontology universally unique identifier**
10	**5, 6, 7**	**DIG text string, a semantic resource description**
11-31	Reserved	

Table 4
Type descriptor values in the proposed modified version of the Bluetooth SDP

Since the communication is referred to the peer layers of the protocol stack, each transaction is represented by one request Protocol Data Unit (PDU) and another PDU as response.

In every SDP PDU, we have a structure like the one pictured in Figure3; the header contains the identifier of the PDU, the identifier of the transaction and the length of the next PDU parameters field.

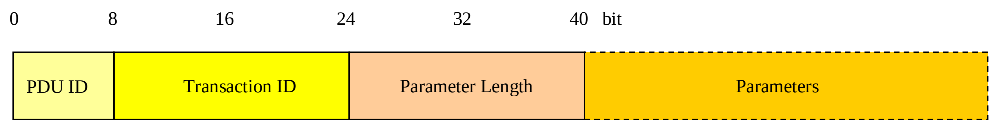

Fig. 3. The Service Discovery Protocol elementary PDU structure

If the SDP request needs more than a single PDU (this case is frequent enough when we use semantic service discovery) the SDP server generates a partial response and the SDP client waits for next part of the complete answer.

By adding two SDP features *SDP_OntologySearch* (request and response) and *SDP_SemanticServiceSearch* (request and response) to the original standard (exploiting not used PDU ID) we are able to introduce, together with the original SDP capabilities, further semantic enabled discovery functionalities.

The transaction between requester and hotspot starts after ad-hoc network creation. When a user queries for a specific service/resource, the generic steps, until response is provided, are detailed in the following, with reference to Figure4. Recall that within the hotspot semantic descriptions of resources are permanently stored. Anyway, further resources could be retrieved from mobile hosts connected via Bluetooth to the hotspot itself.

(i) The user searches for a specific ontology identifier by submitting one or more $OUUID_R$ she manages by means of her client application

(ii) The hotspot selects OUUIDs it copes with, matching each $OUUID_R$, furthermore it forwards the request to mobile clients within the piconet in broadcast

(iii) Each mobile host sends to the hotspot one or more resource description classified by means of the required $OUUID_R$

(iv) The hotspot groups all the collected OUUIDs and replies to the requester

(v) The user sends a service request (R) to the hotspot

(vi) The hotspot extracts descriptions of each resource classified with the previously selected $OUUID_R$, cached within the hotspot itself

(vii) The hotspot performs the matchmaking process between R and selected resources it shares. Taking into account the matchmaking results, all the resources are ranked with respect to R

(viii) The hotspot replies to the user with ranked discovered resources information

We remark that basically all the previous steps are based on the original SDP in Bluetooth. No modifications are made to the original structure of transactions, but simply we differently use the SDP framework.

Table5 shows the overall PDU types in our modified version of the Bluetooth Service Discovery Protocol.

In what follows we outline the structure of the SDP PDUs we added within the original framework to enable a semantic based resource discovery.

The first one is the ***SDP_OntologySearchRequest*** PDU, whose parameters are shown in Table6.

The *OntologySearchPattern* is a data element sequence where each element in the sequence is a OUUID. The sequence must contain at least 1 and at most 12 OUUIDs, as for UUID in the original standard. The obtained list of OUUIDs is an ontology search pattern. The *ContinuationState* parameter maintains the same purpose of the original Bluetooth [6].

The ***SDP_OntologySearchResponse*** PDU is generated by the previous PDU. Table7 shows its parameters.

The *TotalOntologyCount* is an integer containing the number of ontology

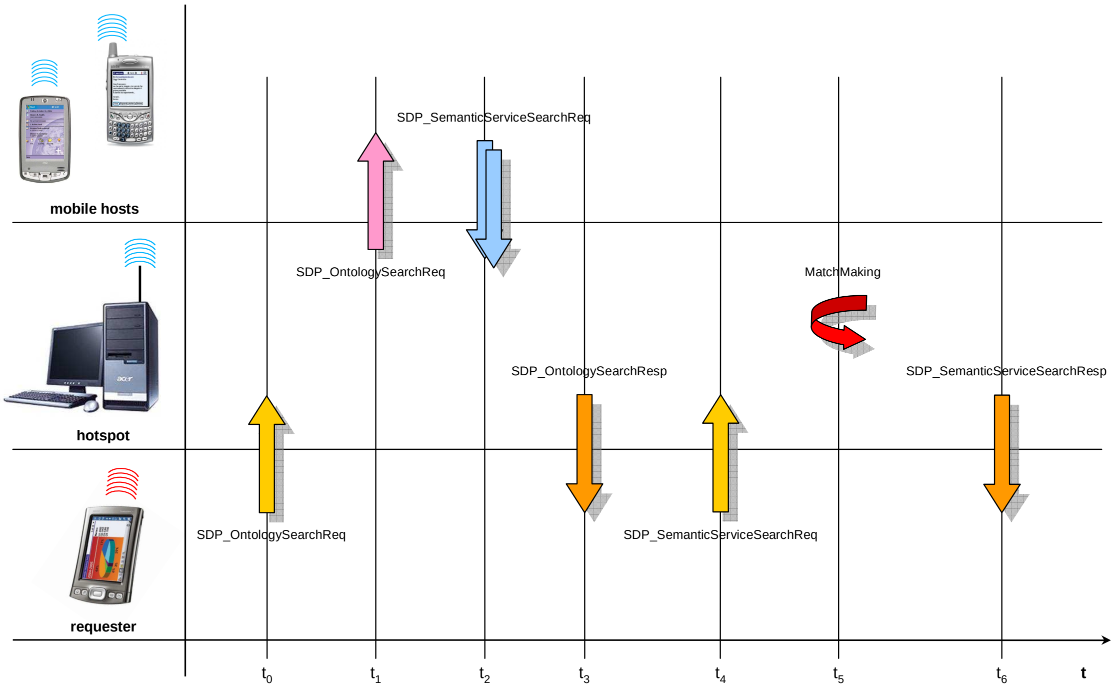

Fig. 4. Client-server interaction within a semantic enabled Bluetooth piconet

PDU ID	DESCRIPTION
0x00	Reserved
0x01	SDP_ErrorResponse
0x02	SDP_ServiceSearchRequest
0x03	SDP_ServiceSearchResponse
0x04	SDP_ServiceAttributeRequest
0x05	SDP_ServiceAttributeResponse
0x06	SDP_ServiceSearchAttributeRequest
0x07	SDP_ServiceSearchAttributeResponse
0x08	**SDP_OntologySearchRequest**
0x09	**SDP_OntologySearchResponse**
0x0A	**SDP_SemanticServiceSearchRequest**
0x0B	**SDP_SemanticServiceSearchResponse**
0x0C-0xFF	Reserved

Table 5
PDU IDs with corresponding descriptions

identifiers matching the requested ontology pattern; the *OntologyRetrieved-Pattern* is a data element sequence where each element in the sequence is a OUUID matching at least one of those sent with the *OntologySearchPattern*. If no OUUID matches the pattern, the *TotalOntologyCount* is set to 0 and

PDU ID	parameters
0x08	**OntologySearchPattern** ContinuationState

Table 6
SDP_OntologySearchRequest PDU parameters

PDU ID	parameters
0x09	**TotalOntologyCount** **OntologyRetrievedPattern** ContinuationState

Table 7
SDP_OntologySearchResponse PDU parameters

the *OntologyRetrievedPattern* contains only a specific reserved OUUID able to allow the browsing by the client of all the OUUIDs managed by the hotspot (see the following ontology browsing mechanism for further details). Hence the pattern sequence contains at least 1 and at most 12 OUUIDs.

The ***SDP_SemanticServiceSearchRequest*** PDU follows *SDP_Ontology SearchResponse* one. Table8 shows its parameters.

PDU ID	parameters
0x0A	**SemanticResourceDescription** **ContextAwareParam1** **ContextAwareParam2** MaximumResourceRecordCount ContinuationState

Table 8
SDP_SemanticServiceSearchRequest PDU parameters

The *SemanticResourceDescription* is a data element text string in DIG formalism describing the resource we are searching for, *ContextAwareParam1* and *ContextAwareParam2* are data element unsigned integers.

In the presented case study, which models a museum scenario, we use only one of them to indicate the time still available for the visitor to tour the museum. The other context-aware parameter is not used in this application and is reserved for future purposes. Since a generic client interacting with a hotspot is in its range, using the above PDU parameter she can impose -among others- a proximity criterion in the resource discovery policy. The other parameters maintain the the original Bluetooth meaning [6].

The ***SDP_SemanticServiceSearchResponse*** PDU is generated by the previous PDU. Its parameters are reported in Table9.

The *SemanticResourceRecordHandleList* includes a list of resource record handles. Each of the handles in the list refers to a resource record potentially

PDU ID	parameters
0x0B	TotalResourceRecordCount CurrentResourceRecordCount **SemanticResourceRecordHandleList** ContinuationState

Table 9
SDP_SemanticServiceSearchResponse PDU parameters

matching the request. Note that this list does not contain header fields, but only the 32-bit record handles. Hence, it does not have the data element format. The list of handles is arranged according to the relevance order of resources, excluding resources not compatible with the request. The other parameters maintain the same purpose of the original Bluetooth [6].

Notice that in all previous cases, the error handling is managed with the same mechanisms of Bluetooth standard [6]. Hence possible faults generated within the previous frame exchange are referred to the error types Bluetooth can cope with.

4.3 Ontology management

Each resource retrieval session starts after settling between client and server the same ontology identifier (OUUID). This is needed to grant a common vocabulary in communications occurring between a mobile client and a resource provider. Nevertheless if a client does not support any ontology or if the supported ontology is not managed by the hotspot, it is desirable to discover what kind of resource class (and then what OUUIDs) are handled by the zone server without any a priori information about resources. For this purpose we use the *service browsing* feature [6] in a slightly different fashion w.r.t. the original Bluetooth standard, so we call this mechanism **ontology browsing**. It is based on an attribute shared by all semantic enabled resource classes, the *BrowseSemanticGroupList* attribute, which contains a list of OUUIDs. Each of them represents the browse group a resource can be associated to.

Groups are organized in a hierarchical fashion, hence when a user wants to browse a hotspot resource class, she can create an ontology search pattern containing the OUUID that represents the root browse semantic group. All resources that may be browsed at the top level are made members of the root browse semantic group, by having the root browse group OUUID as a value within the *BrowseSemanticGroupList* attribute.

Generally, a hotspot such those available in museums, supports relatively few resource classes, hence all of their resources will be placed in the root browse group. However, the resources exposed by a provider may be organized in a browse group hierarchy, by defining additional browse groups below the root browse group.

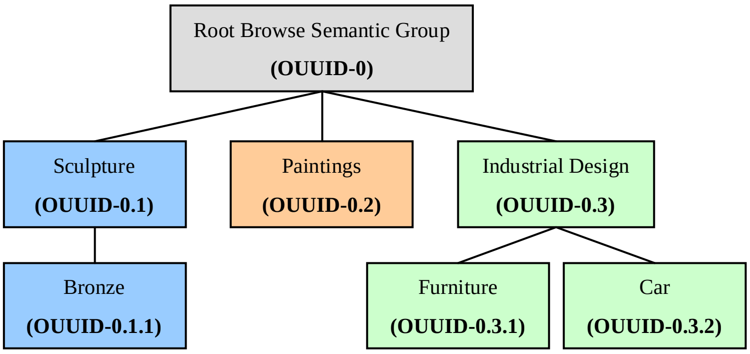

Fig. 5. A simplified ontology browsing hierarchy

Having determined the work of art category (resource class) and the corresponding reference ontology, the client can also download its DIG version from the hotspot as a *.jar* file (such a file extension -among other things- also allows a total compatibility with the Connected Limited Device Configuration (CLDC) technology).

4.4 Additional features

In advanced mobile scenarios, the match between a request and a provided resource involves not only the description of the resource itself but also data-oriented properties. It would be quite strange to have a P2P interaction among mobile hosts in an ubiquitous context without taking into account remaining battery power, memory capabilities, subsidiary equipments of a mobile device, among others. The requester should be able to specify what type of device she owns, its computational capabilities, the remaining battery power. Hence, the overall match value should depend not only on the semantic distance between the description of the demand and of the resource, but also on those subsidiary values. The overall utility function has to combine all these values to give a global value representing the match degree.

Also notice that, in ad-hoc applications, in addition to "instrumental" parameters, also context-aware variables could influence matching results. For example, in our museum case study, we consider the physical distance between requester and resource to properly weigh the match degree. The distance becomes an interesting value since a user has a temporal deadline to conclude her visit. Hence, a resource or a tour within the museum might be preferred also according to its proximity to the user.

We express this distance in terms of time needed to reach the room where a resource is, leaving from the hotspot area. In such a way the hotspot will exclude resources not reachable by the user within the opening hours of museum or within the deadline time for visit. So it will assign to resources unlikely reachable a weight smaller than one assigned to easily reachable ones. The above approach can be further extended to other data-type properties [23].

The utility function we use depends on:

- t_D: time interval available to the client
- t_R: time to reach the resource room and come back up to the exit, leaving from the hotspot area
- *s_match*: score computed during the semantic matchmaking process through *rankPotential* algorithm [13]

$$f(s\_match, t_D, t_R) = \frac{2}{3} s\_match + \frac{1}{3} tanh \frac{t_D - t_R}{\alpha}$$

In the tests we carried out, we found $\alpha = 10$ seems to be in accordance with experience, but it could be changed according to different specific scenarios.

5 Case study

In what follows we detail the discovery and matchmaking process with respect to our case study, deployed in a museum art gallery. Ontologies shared on the web are used to describe its resources, whose related descriptions are also available on the museum web site.

For the sake of simplicity we refer to a scenario such as the one pictured in Figure 6, in order to explain the approach and the rationale behind it. The purpose of such piconet is to find, within the ad-hoc network, services and resources requested by users (with a generic Bluetooth compatible device) searching among on-line available ones.

When a user becomes a member of the ad-hoc network, she is able to ask for a specific service/resource (by submitting a semantic-based description). After receiving a user request, the zone resource provider selects resource descriptions it stores and collects those available in the area, to perform the discovery process for the request. Results are then ranked and returned to the user.

Notice that each resource retrieval session starts after submission from client side to server of the ontology identifier (OUUID) in order to select possible hosts, suitable for requested services. Each host processes the incoming request at SDP layer.

Let us now suppose we are visiting an art gallery, which classifies its works of art in a local Knowledge Base (KB) with a simplified ontology (marked with a specified identifier we indicate $OUUID_M$) reported in Figure 7.

For the sake of simplicity, in the simple ontology in Figure 7 only subclass and disjoint relations are represented.

Let us also suppose that a generic visitor (`user2`) is visiting the same museum and in particular the same room we are in. She is interested in dadaist style and she has previously downloaded on her mobile device a document file

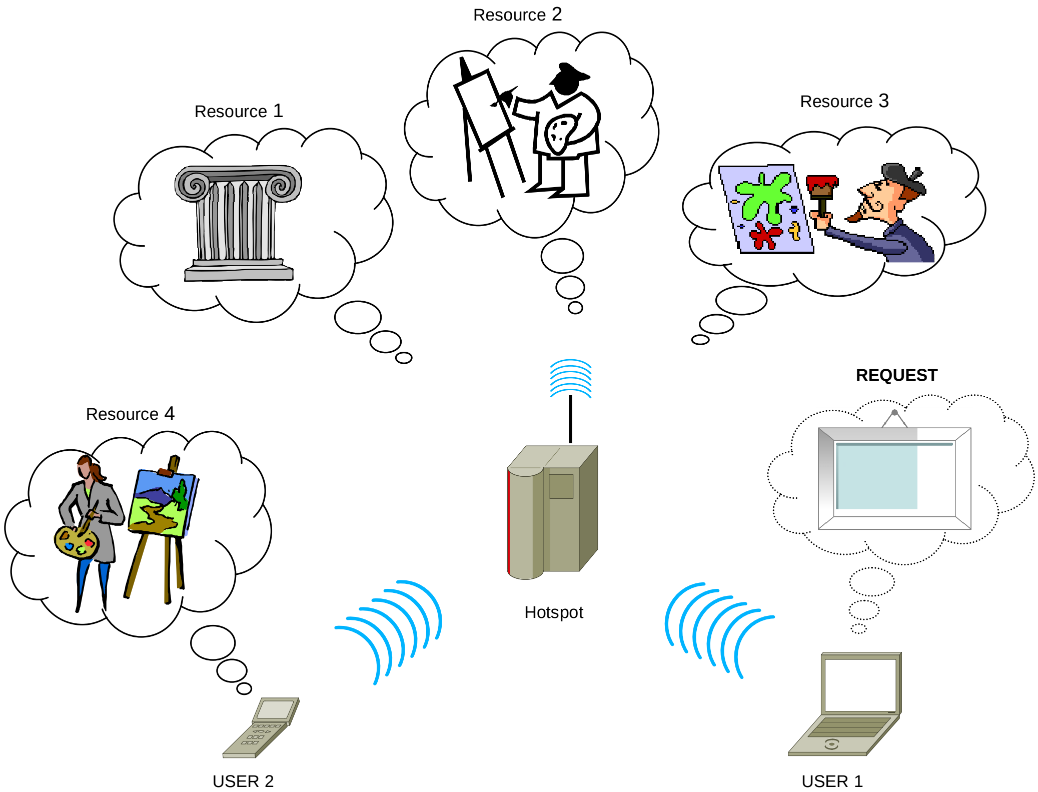

Fig. 6. Semantic based P2P piconet infrastructure

Painting ⊑ WorkOfArt
Sculpture ⊑ WorkOfArt
Dadaism ⊑ Style
Futurism ⊑ Style
Symbolism ⊑ Style
Abstraction ⊑ Subject
Nature ⊑ Subject
Portrait ⊑ Subject
Religious ⊑ Subject
Scenery ⊑ Subject
PaintingTechnique ⊑ Technique
SculptureTechnique ⊑ Technique
Bust ⊑ Sculpture
Statue ⊑ Sculpture
Landscape ⊑ Nature
Seascape ⊑ Nature
StillLife ⊑ Nature
Icon ⊑ Portrait
Naked ⊑ Portrait
SelfPortrait ⊑ Portrait
Icon ⊑ Religious
Cityscape ⊑ Scenery
Landscape ⊑ Scenery
Seascape ⊑ Scenery
Distemper ⊑ PaintingTechnique
Oil ⊑ PaintingTechnique
Pastel ⊑ PaintingTechnique
Watercolors ⊑ PaintingTechnique
Carving ⊑ SculptureTechnique
LowRelief ⊑ SculptureTechnique
Mold ⊑ SculptureTechnique
Sculpture ⊑ ¬Painting
PaintingTechnique ⊑ ¬SculptureTechnique
Style ⊑ ¬Subject
Statue ⊑ ¬Bust
Watercolors ⊑ ¬Oil
Watercolors ⊑ ¬Pastel
Watercolors ⊑ ¬Distemper
Oil ⊑ ¬Pastel
Oil ⊑ ¬Distemper
Pastel ⊑ ¬Distemper
Mold ⊑ ¬Carving
Mold ⊑ ¬LowRelief
Carving ⊑ ¬LowRelief

Fig. 7. A simple art gallery ontology used as reference in the example

with more information concerning dadaist oils on canvas from the Knowledge Base of the consortium the art gallery is associated with. This KB classifies its contents by means of the previous $OUUID_M$ ontology.

Let us imagine that the hotspot KB is populated with the following individuals:

LN *League of nations* classified as:
$\texttt{Painting} \sqcap \forall\texttt{hasStyle.Dadaism} \sqcap$
$\forall\texttt{hasTechnique.Pastel}$

FU *Forme uniche nella continuità di spazio* classified as:
$\texttt{Sculpture} \sqcap \forall\texttt{hasStyle.Futurism}$

MSV *Mont Saint Victor* classified as:
$\texttt{Painting} \sqcap \forall\texttt{hasTechnique.Oil}$

SP *Self-portrait* classified as:
$\texttt{WorkOfArt} \sqcap \forall\texttt{hasSubject.SelfPortrait}$
$\sqcap\forall\texttt{hasStyle.Symbolism}$

WCH *Woman combing her hair* classified as:
$\texttt{Statue} \sqcap \forall\texttt{hasSubject.Naked}$

Moreover the document file on the `user2` PDA is a specific document on painting *Broyeuse de chocolat no.2* classified as: $\texttt{Painting} \sqcap \forall\texttt{hasStyle.Dadaism} \sqcap \forall\texttt{hasTechnique.Oil}$ (we will indicate such resource with **BC**).

Hence we can submit to the hotspot a semantic request for more info on *oil dadaist paintings with landscape as subject*. The request will be formulated in DLs format as $D = \texttt{Painting} \sqcap \forall\texttt{hasStyle.Dadaism} \sqcap \forall\texttt{hasTechnique.Oil} \sqcap \forall\texttt{hasSubject.Landscape}$ with respect to the ontology identified by the $OUUID_M$.

The hotspot will search for $OUUID_M$ and will find the `user2` client resource plus the others already known. In fact `user2` is in the range of the hotspot and she exposes a resource classified by the same ontology managed by the matchmaker.

Then `user2` sends the DIG description of her resource (together with the OUUID) to the hotspot which calls the matchmaker module for rank computation.

In Table10 ranked discovery results are presented. The second column shows whether the resource is compatible or not with D and, in case, the *rankPotential* computed result.

Notice that using only semantic match values (*s_match*), *BC* results the best choice for the demander and *LN* or *MSV* the second ones indifferently. On the other hand, taking into account context-aware information related to physical distance, the order is changed. The ranked list returned by the hotspot is a strict indication for the user about best available resources in the art gallery piconet in order of relevance with respect to the request and to the

resource	compatible (Y/N)	score	s_match	$f(\cdot)$
BC	Y	3	0.484	0.638
LN	Y	4	0.424	0.750
MSV	Y	4	0.424	0.662
SP	Y	9	0.121	0.408
WCH	N	-	-	-
FU	N	-	-	-

Table 10
Matchmaking results

user preferences.

After having selected the best resource, the resource provider will receive a connection request from the user mobile device with its connection parameters and in this manner the transaction may start.

6 Advanced Semantic based Services for the User

In order to promote sharing of resources among hosts in a piconet, additional user-oriented services are provided. These promotional services are integrated within the discovery infrastructure to encourage mobile clients coming into an ubiquitous context, to make available contents they manage, so enriching the context and increasing the interaction level.

6.1 P2P user community

The sharing of a resource is the fundamental requisite to aggregate mobile users in the ad-hoc environment grouping them in communities.

If a user shares a resource, the system will communicate each choice attempt for the resource itself, coming from other clients. The resource owner could decide to establish private communication sessions with one or more users interested in contents she exposes contacting them. Each session could be established thanks to exchange of *calling cards*, containing short information about users like a brief profile as well as contacts.

When a user selects a resource, a calling card is automatically sent to the resource provider. In this manner a host sharing owned resources will receive short info about mobile clients requiring them. This simple mechanism allows to establish a more direct dialog among users interested in a specific subject and furthermore it imposes a credential policy in resource usage.

6.2 Static recommendation service

The prototype we present also allows to implement an elementary recommendation system. If a user enters the hotspot range, and decides to share her resources by providing the system with a description of her offered services, the system itself will return a list with best proposals for her interests. In fact MAMAS-tng on the hotspot will match the semantic description of shared resources with all other available ones in the entire museum area. Hence the zone server will suggest to the user the resources that better match ones she owns.

With respect to the previous case, `user2` could receive a proposal for specific resources by the system, according to her preferences.

In particular the hotspot will match the semantic description of `user2` shared resources with all the ones both available in the art gallery and stored in its database. Hence it will compute the rank value and will suggest to `user2` only most promising results.

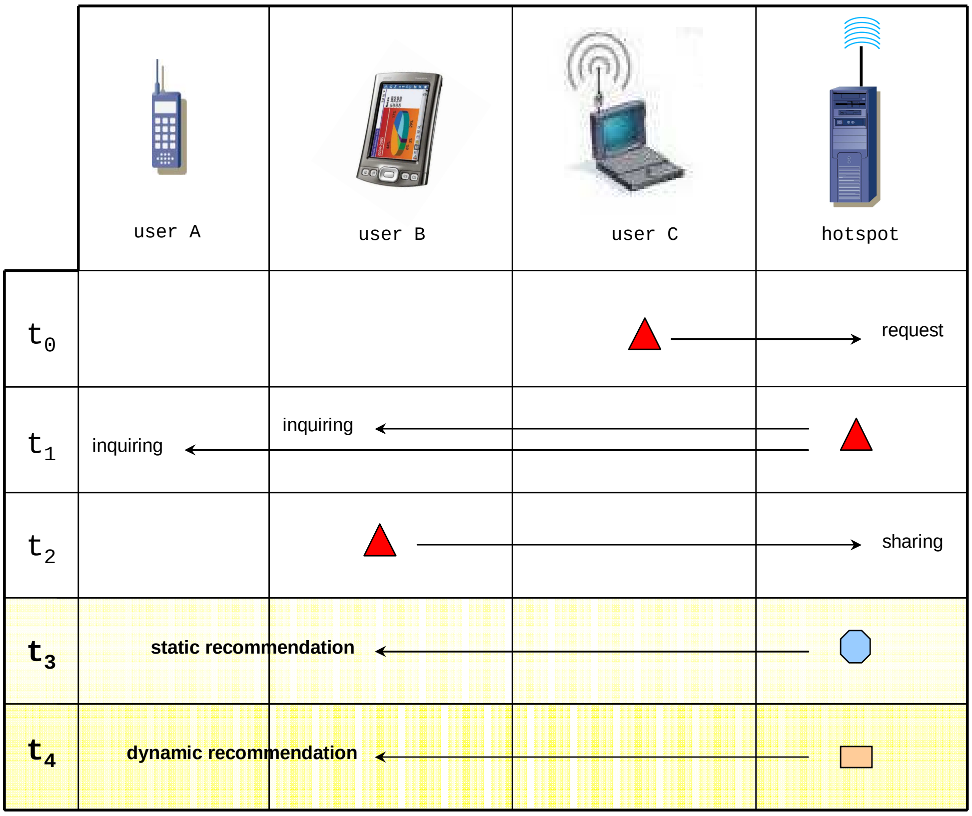

Fig. 8. Static and dynamic recommendation feature

In order to explain this feature, let us suppose `user2` shares her **BC** document file. Consequently the hotspot starts a recommendation process for `user2`. It will match the semantic description of **BC** with all the ones in its database (**LN**, **MSV**, **SP**, **WCH**, **FU**). With respect to our example, Table11 ensues.

The best matching resources are suggested to the user. The system proposes to `user2` to visit rooms where they are located.

resource	compatible (Y/N)	score	location
LN	Y	1	room A - floor 2
MSV	Y	1	room B - floor 2
SP	Y	6	room A - floor 1
WCH	N	-	-
FU	N	-	-

Table 11
Static recommendation results

6.3 Dynamic recommendation service

Previous recommendation feature works in a static fashion. It is basically a suggestion service referred to resources permanently classified within the hotspot.

Nevertheless, recommendation to users can also be done in a dynamic way: the hotspot can suggest to a user possible requests to be submitted to the system. With reference to our case study, it will match the **BC** description against semantic descriptions of other shared resources. After computing the best rank it extract requests submitted by users which shared them.

A simple example will clarify the approach. Let us suppose `user3` and `user4` have just come in the museum. `user3` shares two resources:

r1.3 *Resource n.1 by user3* classified as:
`Bust` ⊓ ∀`hasSubject.Religious`

r2.3 *Resource n.2 by user3* classified as:
`WorkOfArt` ⊓ ∀`hasStyle.Symbolism` ⊓ ∀`hasSubject.Icon`

and submitted the request:

d1.3 *Request n.1 by user3* classified as:
`WorkOfArt` ⊓ ∀`hasSubject.Religious`

On the other hand `user4` shared the resource:

r1.4 *Resource n.1 by user4* classified as:
`Painting` ⊓ ∀`hasStyle.Dadaism`⊓
∀`hasSubject.StillLife`

and submitted the following requests:

d1.4 *Request n.1 by user4* classified as:
`Painting` ⊓ ∀`hasStyle.Dadaism` ⊓ ∀`hasSubject.Nature`

d2.4 *Request n.2 by user4* classified as:
`Painting` ⊓ ∀`hasStyle.Dadaism`⊓
∀`hasTechnique.Distemper`

The system will match the **BC** description from `user2` with **r1.3**, **r2.3** and **r1.4**. Then it will determine the ranked list in Table12.

Hence the system recommends to `user2` requests **d1.4** and **d2.4** submitted to the hotspot by `user4` (observe that **r1.4** resource shared by `user4` best

resource	compatible (Y/N)	score
r1.4	Y	1
r2.3	Y	6
r1.3	N	-

Table 12
Dynamic recommendation results

matches **BC** resource managed by `user2`).

Notice that, even if `user2` does not submit any explicit profile description to the hotspot, the system tries to identify her preferences. Static and dynamic recommendation features allow an elementary and implicit user profiling implementation. Without inserting any profile, the user exposes her interests by means of resources she owns and wishes to share in the ad-hoc virtual community. Then the system proposes to the user a collection of most specific services based on preferences. The system does not force the user to insert her profile for selecting suitable contents, but equally succeeds to extract possible services in her interests.

7 Related Work

There is a widespread request for an increase of discovery features in wireless contexts like Bluetooth piconets. Semantic service discovery via matchmaking in the Bluetooth framework was first investigated in [1]. There the need for discovery mechanisms more powerful than those of the original standard, inadequate for modern ubiquitous scenarios, was clearly pointed out for the first time. The issue of approximate matches in the absence of exact ones was discussed, but no formal framework was given: a logical formulation was expected to devise correct algorithms to classify and rank matches.

In recent years dynamic distributed systems have been developed adopting various technologies and for different purposes. In [11] a Jini-based distributed agent framework is used in a hybrid agent-oriented/service-oriented approach, whereas in [25] semantic user profiles are introduced to increase accuracy in matching services.

In a mobile environment, where subjects on-line are continuously in evolution, modeling real peer to peer interaction calls for a common vocabulary to classify semantic descriptions of services. In fact two or more clients in the piconet who want to share information must have a common way for describing them.

Existing service discovery systems do not support a well defined common ontology infrastructure. Architectures like Jini allow to "capture" the ontology among services by means of mechanisms like Java classes which are difficult to be widely adapted. This limitation, as admitted in [11] and in [10], is due to the lack of shared ontology support. In [10] it is assumed that a client request is described by means of the same ontology a service uses for describing itself. This assumption is fundamental because it restricts the discovery only to services classified in the same manner, but there is no mention to the technique to reach this objective. Here we proposed a simple method for ontology matching prior to service discovery. The preliminary ontology matching grants a quick restriction of the available services only to those semantically suitable.

In [25] a mobile environment is presented where semantic services are matched against semantic user profiles. Here, if there is no intersection between user interests and service offers, authors conclude the user is not interested in the service. A complete and integrated solution for matching degree determination is not available.

In [24] SDP@HA is presented, a system where service discovery is applied to home environments. Appliances are divided into three classes according to their computational capabilities. Such classification imposes to distinguish service discovery protocol functions. Furthermore several assumptions are done about services identification. A *catalogue service* is employed to classify available services, and discovery is limited to identify device type, service type or attributes. No semantic approaches are presented to solve limitations of syntactic device discovery. With respect to SDP on Bluetooth, our approach allows to obtain features similar to communication framework presented in SDP@HA. In fact we use an hybrid client/server architecture in sessions establishment but also peer to peer in contents sharing among hosts. In [24] peer to peer communication occurs in a hardware mode and there is no references to the high level user mode knowledge sharing.

Chen et al. in [11] present an hybrid approach, agent/service oriented, to perform dynamic service discovery in mobile environments based on Bluetooth-like devices. For such purpose authors employ Jini platform and enrich it with a distributed agent layer. In fact Jini Lookup Service does not solve some important service discovery problems. Therefore the provided framework seems to require too large computational resources to be easily adapted to a real mobile scenario. The agent software layer should perform semantic enabled service discovery managing inexact matching. This is still too computationally heavy to run on a mobile device. Hence, as admitted by the authors, a proxy agent which resides in a computer on the wired side is needed, so that handheld devices are responsible only for GUI. Furthermore in [11] there is no mention to the solution of the inexact matching issue. No formal methods to determine approximate matches are outlined. Finally the proposed system is strictly client server. It does not allow to implement a real P2P scenario. The sharing of resources managed by a network client with other mobile hosts is not foreseen, hence it could be obtained only by loading shared services into a local database and by registering them into Jini Lookup Service. This is a significant restriction because it makes not possible a direct communication among two or more peers in the ad-hoc network, by-passing Jini Lookup or any broker agent.

The spontaneous and occasional collaboration among mobile users is investigated in [20]. There an example of collaborative environment is presented, where ontologies are used to infer new information about mobile users profile. In that cooperative context, a matchmaking service communicates with a localization one, which discovers all the MAC addresses of the mobile devices in the environment. Matchmaking service compares the user profiles associated to those MAC. There is not a close merging between discovery phase and matching one. The integration of the proposed matchmaking system in a complex semantic service discovery architecture is still lacking.

[18] introduces a framework for resource retrieval based on a set of self-organized discovery agents which manage a directory information where resources can be searched out by using hash indexing. In addition, the proposed system allows to perform a dynamic selection of best service provider according to supplied QoS. The agents divide the network into domains and collect intra/inter domain QoS information to choose appropriate providers. Unfortunately the proposed framework is based on a purely string matching discovery.

In [17], the concept of context attribute has been defined to extract and subsequently manage information about context during the resource discovery process. As devised in that paper a context attribute could include network or client settings, quality of service parameters as well as other specified variables. Such attributes are dynamically determined and evaluated by the lookup services and contribute to refine the traditional discovery (performed by means of static attributes). Although this is an improvement w.r.t. syntactic resource discovery, a complete and comprehensive framework to support context awareness should be provided.

As investigated in [26] a significant application of semantic ad-hoc networks can be made just in tourism, where a more efficient system is desirable for searching, delivering and sharing information. After *U-commerce*, where the use of ubiquitous computing supports personalized transactions among companies and buyers, *U-tourism* has become a new perspective of tourism. In spite of lack of specific technologies to support tourists, there is a widespread interest for personalized virtual guides. [26] presents an articulated proposal to solve such question, but it is exclusively addressed to tourist purposes, and the proposed architecture does not appear suitable in different scenarios.

8 Conclusion and Future Work

In this paper we have exploited Knowledge Representation techniques and technologies to enrich the capabilities of the Service Discovery Protocol in Bluetooth. Adding information modeled using languages with a well-defined formal semantics, we have made the SDP able to manage also non-exact matches between the requested service/resource and the offered ones. In such a way a semantic layer has been integrated in the existing standard Service Discovery Protocol for Bluetooth, allowing a semantic-based discovery and ranking process.

The approach extends the basic SDP to discover and rank both resources and advanced peer services (as for the recommendation and community features), by adding reasoning mechanisms to the discovery process. It is noteworthy that this enrichment does not require any modification to the standard Bluetooth, thus allowing a smooth coexistence of semantic discovery with syntactic one.

The framework, originally developed within the CNOSSO project –New Technologies for Fruition and Development of Cultural Goods–, adopts these enhanced features and uses them to provide various user-oriented services, which benefit of a Knowledge Representation approach. In particular, beyond classical P2P semantically-enabled resource sharing, also here semantic-based recommendation system and community formation have been presented.

Under development is an integrated push service to enable ontology support in ubiquitous contexts.

References

[1] S. Avancha, A. Joshi, and T. Finin. Enhanced Service Discovery in Bluetooth. *IEEE Computer*, 35(6):96–99, June 2002.

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

[3] F. Baader, I. Horrocks, and U. Sattler. Description Logics as Ontology Languages for the Semantic Web. *Lecture Notes in Artificial Intelligence*, Springer, 2003.

[4] M. Barbeau. Service Discovery Protocols for Ad Hoc Networking. In *Workshop on Ad-hoc Communications (CASCON '00)*, 2000.

[5] S. Bechhofer, R. Möller, and P. Crowther. The DIG Description Logic Interface. In *DL '03*. CEUR, September 2003.

[6] Bluetooth. http://www.bluetooth.com.

[7] A. Borgida. Description Logics in Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):671–682.

[8] R. Brachman and H. Levesque. The Tractability of Subsumption in Frame-based Description Languages. In *4th National Conference on Artificial Intelligence (AAAI-84)*, pages 34–37. Morgan Kaufmann, 1984.

[9] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the Decidability of Query Containment under Constraints. In *17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '98)*, pages 149–158. ACM Press, 1998.

[10] D. Chakraborty, F. Perich, S. Avancha, and A. Joshi. DReggie: Semantic Service Discovery for M-Commerce Applications. In *Workshop on Reliable and Secure Applications in Mobile Environment*, 2001.

[11] H. Chen, A. Joshi, and T. Finin. Dynamic Service Discovery for Mobile Computing: Intelligent Agents Meet Jini in the Aether. *Cluster Computing*, 4(4):343–354, 2001.

[12] DAML+OIL specifications. http://www.daml.org/2001/03/daml+oil-index.html.

[13] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A System for Principled Matchmaking in an Electronic Marketplace. *International Journal of Electronic Commerce*, 8(4):9–37, 2004.

[14] F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. In G. Brewka, editor, *Principles of Knowledge Representation: Studies in Logic, Language and Information*, pages 191–236. CSLI Publications, 1996.

[15] OWL-S: Semantic Markup for Web Services. http://www.daml.org/services/owl-s/1.1/overview/.

[16] E.A. Gryazin. Service Discovery in Bluetooth. http://www.hpl.hp.com/techreports/2002/HPL-2002-233.pdf.

[17] C. Lee and S. Helal. Context Attributes: An Approach to Enable Context-awareness for Service Discovery. In *Symposium on Applications and the Internet (SAINT '03)*, pages 22–30, 2003.

[18] J. Liu, Q. Zhang, B. Li, W. Zhu, and J. Zhang. A Unified Framework for Resource Discovery and QoS-Aware Provider Selection in Ad Hoc Networks. *ACM Mobile Computing and Communications Review*, 6(1):13–21, January 2002.

[19] D. McGuinness, R. Fikes, J. Hendler, and L. Stein. DAML+OIL: An Ontology Language for the Semantic Web. *IEEE Intelligent Systems*, 17(5):72–80, 2002.

[20] R. Prestes, G. Carvalho, R. Paes, C. Lucena, and M. Endler. Applying Ontologies in Open Mobile Systems. In *Workshop on Building Software for Pervasive Computing (OOPSLA '04)*, 2004.

[21] RDF Primer-W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-primer/.

[22] M. Ruta, T. Di Noia, E. Di Sciascio, and F.M. Donini. Semantic enabled Resource Discovery, Composition and Substitution in Pervasive Environments. In *13th IEEE MELECON 2006*, May 2006. To appear.

[23] M. Ruta, T. Di Noia, E. Di Sciascio, F.M. Donini, and G. Piscitelli. Advanced Resource Discovery Protocol for Semantic-enabled M-commerce. In *Encyclopedia of Mobile Computing and Commerce (EMCC)*. Idea Group, USA, 2006. To appear.

[24] V. Sundramoorthy, H. Scholten, P. Jansen, and P. Hartel. Service Discovery at Home. In *4th IEEE Pacific-Rim Conf. On Multimedia*, 2003.

[25] A. von Hessling, T. Kleemann, and A. Sinner. Semantic User Profiles and their Applications in a Mobile Environment. In *Workshop on Artificial Intelligence in Mobile Systems (AIMS '04)*, September 2004.

[26] R. Watson, S. Akselsen, E. Monod, and L. Pitt. The Open Tourism Consortium: Laying The Foundations for the Future of Tourism. *European Management Journal*, 22(3):315–326, 2004.