Ontology driven resource discovery in a Bluetooth based m-marketplace

Michele Ruta
Politecnico di Bari
via Re David 200
I-70125 Bari, Italy
Email: m.ruta@poliba.it

Tommaso Di Noia
Politecnico di Bari
via Re David 200
I-70125 Bari, Italy
Email: t.dinoia@poliba.it

Eugenio Di Sciascio
Politecnico di Bari
via Re David 200
I-70125 Bari, Italy
Email: disciascio@poliba.it

Giacomo Piscitelli
Politecnico di Bari
via Re David 200
I-70125 Bari, Italy
Email: piscitel@poliba.it

***Abstract*— We present a semantic-based approach to resource retrieval in an m-commerce scenario. We enhance the original Bluetooth Service Discovery Protocol by integrating a "semantic layer" within the application level of the standard. Given a user request, this layer makes possible a matchmaking process exploiting the semantics of the resources descriptions exposed by a hotspot. It computes the match degree between the request and the goods available in the m-marketplace taking into account both their ontology-based descriptions and "classical" attributes such as price difference, availability, quantity.**

I. INTRODUCTION

New mobile architectures allow for stable networked links from almost everywhere, and more and more people make use of information resources for work and business purposes on mobile systems. Although technological improvements in the standardization processes proceed rapidly, many challenges, mostly aimed at the deployment of value added services on mobile platforms, are still unsolved. In particular the evolution of wireless-enabled handheld devices and their capillary diffusion have increased the need for more sophisticated Service Discovery Protocols (SDPs).

Here we present an approach which enhances Bluetooth SDP, to provide m-commerce resources to the users within a piconet, extending the basic service discovery with semantic capabilities. In particular we exploit and enhance the SDP in order to identify generic resources rather than only services.

We have integrated a "semantic layer" within the application level of the standard Bluetooth stack in order to enable a simple interchange of semantically annotated information between a mobile client performing a query and a server exposing available resources.

We adopt a simple piconet configuration where a stable networked zone server, equipped with a Bluetooth interface, collects requests from mobile clients and hosts a semantic facilitator to match requests with resources available in the m-marketplace. Both requests and resources are expressed as semantically annotated descriptions, so that a semantic distance can be computed as part of the ranking function, to choose most promising resources for a given request.

The rest of the paper is organized as follows: in Section II we briefly introduce the state of the art on service discovery in wireless ad-hoc contexts; in Section III we explain our approach to the problem; in Section IV a case study is presented to clarify the proposed framework; in Section V we comment on related work and in the last SectionVI we expose conclusion and future work.

II. STATE OF THE ART

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 [1].

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 [2].

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, and 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 [3].

Actually there are several issues that restrain the expansion of advanced wireless applications. Among them, the variability of scenarios. An ad-hoc environment is based on short range, low power technologies like Bluetooth [4], which grant the peer to peer interaction among hosts. In such a mobile infrastructure there could be one or more devices providing and using resources 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 as the ones related to electronic commerce. In fact in these cases there is the need to submit articulate requests to the system, to obtain adequate responses [5].

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. In other words Bluetooth SDP is code-based and consequently it can handle only exact matches. Yet, if we want to search and retrieve resources whose description cannot be classified within a rigid schema (e.g. the description of goods in a shopping mall) a more powerful discovery architecture is needed [6]. SDP should be able to cope with non-exact matches [5], and to provide a ranked list of discovered resources, computing a distance between each retrieved resource and the request 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 [7].

In a semantic-enabled Web –what is known as the Semantic Web vision– each available resource should be annotated using RDF [8] with respect to an OWL ontology [9]. There is a close relation between the OWL-DL subset of OWL and Description Logics (DLs) [10] 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 [11] 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.

III. The Proposed Framework

In what follows we outline our framework and sketch the rationale behind it. We adopt a mobile commerce context as reference scenario.

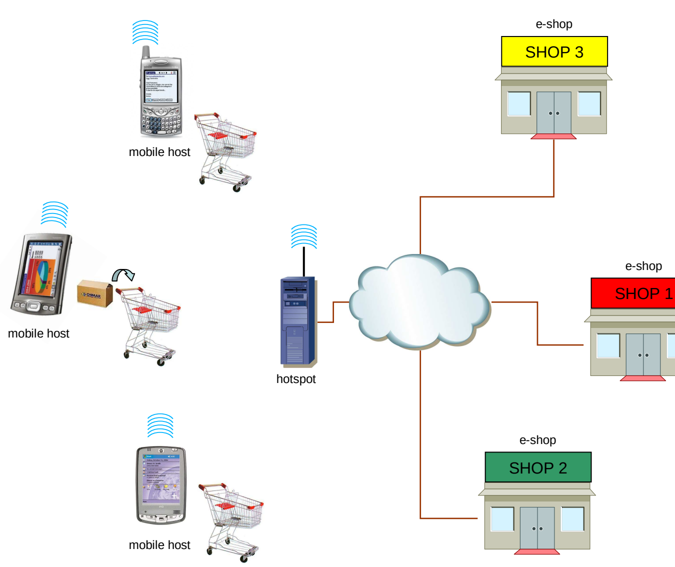

Fig. 1. A simple sketch of the proposed infrastructure

A. Infrastructure

In our mobile environment, a user contacts via Bluetooth a zone resource provider (from now on *hotspot*) and submits her semantically annotated request in DIG formalism. We assume the zone server –which classifies resource contents by means of an OWL ontology– has previously identified shopping malls willing to promote their goods and it has already collected semantically annotated descriptions of goods. Each resource in the m-marketplace owns an URI and is annotated by its OWL description.

The hotspot is endowed with a MatchMaker (in our system we adapt the *MAMAS-tng* reasoner [12]), 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 demand, ranked according to their degree of correspondence to the demand itself.

B. SDP modifications

We allow 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, slightly modifying the existing Bluetooth discovery protocol.

SDP uses a simple request/response method for data exchange between SDP client and SDP server [13]. 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 exclusion of supply descriptions that do not refer to the same ontology of the request [3]. With OUUID matching we do not identify a single service, but directly the context of resources we are looking for, which can be seen as a class of similar services. Each semantically annotated resource is stored within the hotspot as a resource record. A 32-bit identifier is uniquely associated to a semantic resource record within the hotspot, we call *SemanticResourceRecordHandle*.

Each resource 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, we have *ResourceName*, *ResourceDescription*, and a variable number of *ResourceUtilityAttr_i* attributes (in our current implementation 2 of them). *ResourceName* 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 can be associated to context-aware attributes of a resource [14]; in the current implementation we adopt, for example, the price and the physical distance the resource has from the hotspot (expressed in meters or in terms of needed time to get to the resource). We use them as parameters of the overall utility function used to evaluate matchmaking results.

OUUID	ResourceName	ResourceDescription	ResourceUtilityAttr1	ResourceUtilityAttr2	ResourceAttr6	...	ResourceAttrN

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

To allow the representation and the identification of a semantic resource description we introduced in the data representation of the original Bluetooth standard two new data element type descriptor [4]. Resulting types are shown in TableI.

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 I
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, the structure is the one depicted in Figure3.

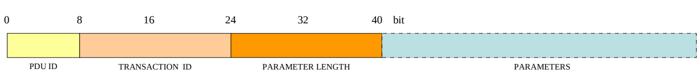

Fig. 3. Standard Bluetooth SDP PDU

If the SDP request needs more than a single PDU (this case is frequent enough if 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 inserted together with the original SDP capabilities further semantic enabled resource search functions.

The transaction between service requester and hotspot starts after ad-hoc network creation. When a user becomes a member of a MANET, she is able to ask for a specific service/resource (by submitting a semantic-based description). The generic steps, up to response is provided, for a service request are detailed in the following, with reference to Figure4.

1) The user searches for a specific ontology identifier by submitting one or more $OUUID_R$ she manages by means of her client application
2) The hotspot selects OUUIDs matching each $OUUID_R$ and replies to the client
3) The user sends a service request (R) to the hotspot
4) The hotspot extracts descriptions of each resource cached within the hotspot itself, which is classified with the previously selected $OUUID_R$
5) 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
6) The hotspot replies to the user

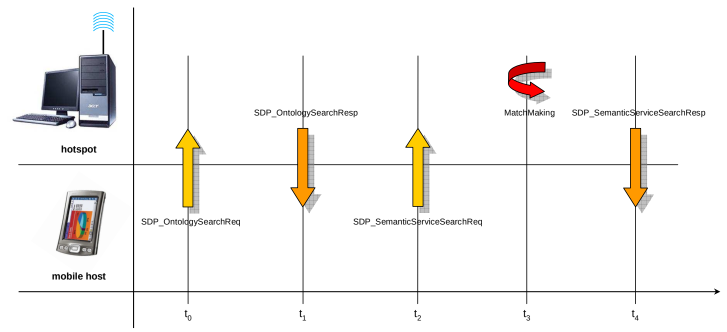

Fig. 4. Client-server interaction within a semantic enabled Bluetooth piconet

It is important to 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.

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

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 II
PDU IDS WITH CORRESPONDING DESCRIPTIONS

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

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

PDU ID	parameters
0x08	**OntologySearchPattern** ContinuationState

TABLE III

SDP_OntologySearchRequest PDU PARAMETERS

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 in the original standard. The list of OUUIDs is an ontology search pattern. The *ContinuationState* parameter maintains the same purpose of the original Bluetooth [4].

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

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

TABLE IV

SDP_OntologySearchResponse PDU PARAMETERS

The *TotalOntologyCount* is an integer containing the number of ontology identifiers matching the requested ontology pattern. Whereas the *OntologyRetrievedPattern* is a data element sequence where each element in the sequence is a OUUID matching at least one sent with the *OntologySearchPattern*. If no OUUID matches the pattern, the *TotalOntologyCount* is set to 0 and the *OntologyRetrievedPattern* contains only a specific 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_OntologySearchResponse* one. TableV shows its parameters.

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

TABLE V

SDP_SemanticServiceSearchRequest PDU PARAMETERS

The *SemanticResourceDescription* is a data element text string in DIG formalism representing the resource we are searching for, *ContextAwareParam1* and *ContextAwareParam2* are data element unsigned integers. In our case study, which models an m-marketplace in an airport terminal, we use them respectively to indicate a reference price for the resource and the hour of the scheduled departure of the flight. 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 [4].

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

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

TABLE VI

SDP_SemanticServiceSearchResponse PDU PARAMETERS

The *SemanticResourceRecordHandleList* includes a list of resource record handles. Each of the handles in the list refers to a resource record potentially 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 [4].

In all the previous cases, the error handling is managed with the same mechanisms and techniques of Bluetooth standard [4].

C. Ontology management

Notice that each resource retrieval session starts after settling between client and server the same ontology identifier (OUUID). 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 merchandise 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 [4] in a slightly different fashion w.r.t. the original Bluetooth standard, so calling 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 may be associated to for browsing.

Groups are organized in a hierarchical fashion, hence when a client desires to browse a hotspot merchandise 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 supports relatively few merchandise classes, hence all of their resources will be placed in the root browse group. However, the resources exposed by a provider may be organised in a browse group hierarchy, by defining additional browse groups below the root browse group.

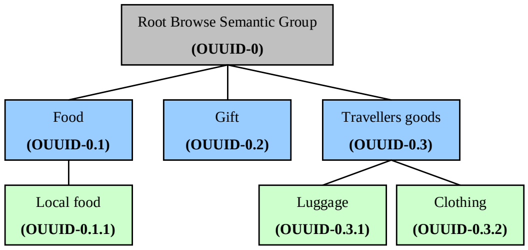

Fig. 5. A simplified ontology browsing hierarchy

Having determined the goods category and the corresponding reference ontology, the client can also download its DIG version from the hotspot as *.jar* file.

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 (for example in the web site of a shopping mall). That is, there is no need for different customized resource descriptions and modelling, if the user employs different applications either on the web or in mobile systems. The formal semantics of the descriptions is unique with respect to the reference ontology and can be shared among different environments.

D. Additional features

In e-commerce scenarios, the match between demand and supply involves not only the description of the good but also data-oriented properties. It would be quite strange to have a commercial transaction without taking into account price, quantity, availability among others. The demander usually specifies how much she is willing to pay, how many items she wants to buy, the delivery date. Hence, the overall match value depends not only on the distance between the (semantic-enabled) description of the demand and of the supply, but also price, quantity, delivery time differences. The overall utility function combines all these values to give a global value representing the match degree.

Also notice that, in m-commerce applications, in addition to "commercial" parameters also context-aware variables should influence matching results. For example, in our airport case study, we consider the price difference but also the physical distance between requester and seller to weigh the match degree. The distance becomes an interesting value since a user has a temporal deadline for shopping: the scheduled time of her flight. Hence, a resource might be chosen also according to its proximity to the user.

We will express this distance in terms of time to elapse for reaching the shop where a resource is, leaving from the hotspot area. In such a way the hotspot will exclude resources not reachable by the user while she is waiting for boarding and 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.

The utility function we used depends on:

- p_D: price specified by the demander
- p_O: price specified by the supplier
- t_D: time interval available to the client
- t_O: time to reach the supplier and come back, leaving from the hotspot area
- s_match: score computed during the semantic matchmaking process through *rankPotential* algorithm [12]

$$f(s\_match, p_D, p_0, t_D, t_O) = \frac{s\_match}{2} + tanh\frac{\frac{t_D - t_O}{\beta}}{3} + \frac{(1+\alpha)p_D - p_O}{6(1+\alpha)p_D}$$

Notice that p_D is weighted by a $(1 + \alpha)$ factor. The idea behind this weight is that, usually, the demander is willing to pay up to some more than what she originally specified on condition that she finds the requested item, or something very similar. In the tests we carried out, we find $\alpha = 0.1$ and $\beta = 10$ are values in accordance with user preferences. These values seem to be in some accordance with experience, but they could be changed according to different specific scenarios.

IV. CASE STUDY

A simple example can clarify the rationale of our setting. Here we will present a case study analogous to the one presented in [6] and we face it by means of our approach.

Let us suppose a user is in a duty free area of an airport, she is waiting for her flight to come back home and she is equipped with a wireless-enabled PDA. She forgot to buy a present for her beloved little nephew and now she wants to purchase it from one of the airport gift stores. In particular she is searching for a learning toy strictly suitable for a kid (she dislikes a child toy or a baby toy) and possibly the toy should not have any electric power supply.

Clearly this request is too complex to be expressed by means of standard UUID Bluetooth SDP mechanism. In addition, non-exact matches between resource request and offered ones is highly probable and the on/off matching system provided by the original standard in this case could be largely inefficient.

The semantic resource request is expressed in a DIG statement exploiting DL semantics and encapsulated in an SDP PDU.

The hotspot equipped with MAMAS reasoner collects the request and initially selects supplies expressed by means of the same ontology shared with the requester. Hence a primary selection of suitable resources is performed. In addition, the matchmaker carries out the matchmaking process between each offered resource in the m-marketplace and the requested one computing a "semantic distance" (*s_match*) [15]. Finally the matchmaking results are ranked and returned to the user.

A subset of the ontology used as a reference in the examples is reported in Figure6. For the sake of simplicity, only the class hierarchy and disjoint relations are represented.

Let us suppose that after the hotspot selects supplies, its Knowledge Base is populated with the following individuals whose description is represented using DL formalism:

- ***Alice_in_wonderland***. Price 20\$. 5 min from the hotspot: $book \sqcap \forall hasGenre.fantasy$
- ***Barbie_car***. Price 80\$. 10 min from the hotspot: $car \sqcap \forall suggestedFor.girl \sqcap \forall hasPowerType.battery$

- ***classic_guitar***. Price 90$. 17 min from the hotspot: $musicalInstrument \sqcap \forall suitableFor.kid \sqcap (\leq 0\ hasPowerType)$
- ***shape_order***. Price 40$. 15 min from the hotspot: $educationalTool \sqcap \forall suitableFor.child \sqcap \forall stimulatesToLearn.shapesAndColors$
- ***Playstation***. Price 160$. 28 min from the hotspot: $video\_game \sqcap \forall hasPowerType.DC$
- ***Winnie_the_pooh***. Price 30$. 15 min from the hotspot: $teddy\_bear \sqcap \forall suitableFor.baby$

The request ***D*** submitted to the system by the user can be formalized in DL syntax as follows:
$learningToy \sqcap \forall suggestedFor.boy \sqcap \forall suitableFor.kid \sqcap (\leq 0\ hasPowerType)$

In addition she sets a reference price of 200$ ($p_D = 200$) as well as the scheduled departure time within 30 minutes ($t_D = 30$).

$battery \sqsubseteq powerSupply$	$music \sqsubseteq discipline$
$DC \sqsubseteq powerSupply$	$shapesAndColors \sqsubseteq discipline$
$teddyBear \sqsubseteq peluche$	$adventure \sqsubseteq genre$
$peluche \sqsubseteq toy$	$fantasy \sqsubseteq genre$
$babyToy \sqsubseteq toy$	$historical \sqsubseteq genre$
$videoGame \sqsubseteq kidToy$	$baby \sqsubseteq person$
$kidToy \sqsubseteq toy$	$boy \sqsubseteq person$
$childToy \sqsubseteq toy$	$child \sqsubseteq person$
$femaleToy \sqsubseteq toy$	$girl \sqsubseteq person$
$piano \sqsubseteq musicalInstrument$	$kid \sqsubseteq person$
$musicalInstrument \sqsubseteq learningToy$	$battery \sqsubseteq \neg DC$
$educationalTool \sqsubseteq learningToy$	$book \sqsubseteq \neg toy$
$learningToy \sqsubseteq toy$	$babyToy \sqsubseteq \neg kidToy$
$maleToy \sqsubseteq toy$	$kidToy \sqsubseteq \neg childToy$
$trainSet \sqsubseteq vehicle$	$babyToy \sqsubseteq \neg childToy$
$car \sqsubseteq vehicle$	$male \sqsubseteq \neg female$
$vehicle \sqsubseteq toy$	$baby \sqsubseteq \neg child$
$male \sqsubseteq sex$	$child \sqsubseteq \neg kid$
$female \sqsubseteq sex$	$baby \sqsubseteq \neg kid$

Fig. 6. A simple toy ontology used as reference in the example

In TableVII matchmaking results are presented. The second column shows whether each retrieved resource is compatible or not with request D and, in case, the *rankPotential* computed result. In the fourth column, matchmaking results are also expressed in a relative form between 0 and 1 to allow a more immediate semantic comparison among requests and different resources and to put in a direct correspondence various rank values.

Finally in the last column results of the overall utility function application are shown.

demand/supply	compatibility (Y/N)	score	s_match	$f(\cdot)$
D/Alice_in_wonderland	N	-	-	-
D/Barbie_car	Y	7	0,364	0,609
D/classic_guitar	Y	3	0,727	0,748
D/shape_order	N	-	-	-
D/Playstation	Y	5	0,546	0,378
D/Winnie_the_pooh	N	-	-	-

TABLE VII

MATCHMAKING RESULTS

Notice that using only semantic match values (*s_match*) *Barbie_car* results the second best choice for the demander and *Playstation* the third one. On the other hand, taking into account context-aware information related to price and physical distance, the order is inverted. The ranked list returned by the hotspot is a strict indication for the user about best available resources in the airport duty free piconet in order of relevance w.r.t. the request. Nevertheless a user can choose or not a resource according to her personal preferences and her initial purposes.

After having selected the best resource, the server of the chosen virtual shop will receive a connection request from the user mobile device with its connection parameters and in this manner the transaction may start. The user can provide her credit card credentials, so that when she reaches the store, her gift will be already packed. This final part of the application is not yet implemented, but it is trivially achievable exploiting the above SDP infrastructure.

V. RELATED WORK

Many proposals about employment of semantics in ubiquitous environments have been devised. Nevertheless existing semantic enabled service discovery architectures generally have some limitations which make them unsuitable for an extensive use in ad-hoc networks.

In [6] w.r.t. the Bluetooth piconets, 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. In this paper the need of managing approximate matches (in the absence of exact ones) was discussed, but no formal framework was given. Furthermore there was no mention to the solution of inexact matching problem and no formal methods to classify rank matches were outlined.

In [16] a Jini-based distributed agent framework is used in a hybrid agent-oriented/service-oriented approach. The authors present a framework to perform a dynamic service discovery in Bluetooth-based mobile environments. They use a Jini platform and enrich it with a distributed agent layer. In fact the simple Jini Lookup Service does not solve some important service discovery problems. The agent layer should perform a semantic based service discovery with inexact matching management. However this feature is computationally heavy to run on a mobile device. Hence, as the authors admit, a proxy agent which resides in a computer on the wired side is needed. The provided framework is too complex to be easily adapted to a really mobile scenario.

In a mobile environment, where subjects on-line are continuously in evolution, to model really suitable applications, first of all a common vocabulary to classify semantic descriptions of resources is required. In fact two or more subjects in a piconet who want to exchange information, must have a common way for describing them. Nevertheless in many cases proposed service discovery systems do not support a well defined common ontology infrastructure.

For example architectures like Jini allow to "capture" the ontology shared among services by means of mechanisms like Java classes which are difficult to be widely adapted.

This limitation, as admitted in [16] and in [3], is due to the lack of a shared ontology support. In particular in [3] 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 within the same class, but there is no mention to the technique to achieve this objective.

A different perspective about resource discovery is introduced in [17]. A mobile environment is presented where semantically annotated services are matched against semantic user profiles. If there is no intersection between user interests and service offers, the authors conclude the user is not interested in the service. A complete and integrated solution for matching degree determination is absent.

The spontaneous and occasional collaboration among mobile users is investigated in [18]. A collaborative context is described where a matchmaking service communicates with a localization service which discovers all the MAC addresses of the mobile devices in the environment. Hence a matchmaking service compares the user profiles associated to those MAC. In this case there is not a close integration between discovery phase and matching phase. A matchmaking system merged with a complex semantic service discovery architecture is still lacking and the ontology support is only used to infer new information about mobile users profiles.

In [19] an application of mobile service discovery in a home context is presented. Home 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 catalog service is employed for available services classifications and the discovery is limited to identification of device type, service type or services attributes. No semantic approaches are presented to solve limitations of syntactic device discovery. In addition peer to peer communications only occur in a hardware mode and there is no references to a possible high level user mode knowledge sharing.

[2] 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 text matching discovery.

In [14], the concept of context attribute has been defined to extract and subsequently manage information about context during the resource discovery process. As devised in the paper a context attribute could include network or client settlings, 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 organic framework to support context awareness is expected.

VI. Conclusion and Future Work

We presented an advanced semantic enabled resource discovery protocol for m-commerce applications. The proposed approach aims to completely reuse and extend the basic functionalities of the original Bluetooth Service Discovery Protocol by adding semantic capabilities to the classic SDP ones. The approach exploits semantic descriptions of discovered resources to model a semantic m-commerce framework.

Future work on the proposed framework aims to the creation of a more advanced DSS able to help a user in a generic m-marketplace. Under development is the support to creation of P2P small communities of mobile hosts where goods and resources are advertised and opinions about shopping are exchanged [20]. Another future activity focuses on strict control of the good advertising. In an m-marketplace, the system will send to various potential buyers best proposals about their interests. We intend to implement a mechanism to advertise goods or services in a more direct and personalized fashion. From this point of view, an additional feature of the system is oriented to the user profiling extraction and management [18] [7] [17]. Without imposing any explicit profile submission to the user, the system could collect her preferences by means of previously submitted requests [7], *i.e.*, by means of the "history" of the user in the m-marketplace.

References

[1] M. Barbeau, "Service discovery protocols for ad hoc networking," in *Workshop on Ad-hoc Communications (CASCON '00)*, 2000.

[2] 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*, vol. 6, no. 1, pp. 13–21, January 2002.

[3] 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.

[4] Bluetooth, "http://www.bluetooth.com."

[5] D. Chakraborty and H. Chen, "Service discovery in the future for mobile commerce," *ACM Crossroads*, vol. 7, no. 2, pp. 18–24, December 2000.

[6] S. Avancha, A. Joshi, and T. Finin, "Enhanced service discovery in bluetooth," *IEEE Computer*, vol. 35, no. 6, pp. 96–99, June 2002.

[7] M. Ruta, T. Di Noia, E. Di Sciascio, F. Donini, and G. Piscitelli, "Semantic based collaborative p2p in ubiquitous computing," in *WI '05 Web Intelligence*. IEEE/WIC/ACM, 2005.

[8] "Rdf primer-w3c recommendation 10 february 2004," http://www.w3.org/TR/rdf-primer/.

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

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

[11] S. Bechhofer, R. Möller, and P. Crowther, "The dig description logic interface," in *DL '03*. CEUR, September 2003.

[12] T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello, "A system for principled matchmaking in an electronic marketplace," *International Journal of Electronic Commerce*, vol. 8, no. 4, pp. 9–37, 2004.

[13] E. Gryazin, "Service discovery in bluetooth," http://www.hpl.hp.com/techreports/2002/HPL-2002-233.pdf.

[14] 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)*, 2003, pp. 22–30.

[15] 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*, vol. 4, no. 4, pp. 345–361, 2005.
[16] H. Chen, A. Joshi, and T. Finin, "Dynamic service discovery for mobile computing: Intelligent agents meet Jini in the aether," *Cluster Computing*, vol. 4, no. 4, pp. 343–354, 2001.
[17] 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.
[18] 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.
[19] V. Sundramoorthy, H. Scholten, P. Jansen, and P. Hartel, "Service discovery at home," in *4th IEEE Pacific-Rim Conf. On Multimedia*, 2003.
[20] S. Avancha, P. D'Souza, F. Perich, and Y. Yesha, "P2p m-commerce in pervasive environments," *ACM SIGecom Exchanges*, vol. 3, no. 4, pp. 1–9, January 2003.