Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications

M. Ruta*, T. Di Noia, E. Di Sciascio

SisInfLab–Politecnico di Bari,
via Re David 200, I-70125 Bari, Italy
E-mail: {m.ruta, t.dinoia, disciascio}@poliba.it
*Corresponding author

F. M. Donini

Università della Tuscia,
via San Carlo 32, I-01100 Viterbo, Italy
E-mail: donini@unitus.it

Abstract: We present a novel resource discovery framework for m-commerce. We extend the original Bluetooth Service Discovery Protocol by integrating a semantic layer within the application level of the standard.

Given a request, this layer makes possible an enhanced discovery process exploiting the semantics of the resources descriptions exposed by a hotspot. The enhancement is compatible with the basic discovery protocol, thus allowing the smooth coexistence of the service discovery approaches.

We present and motivate our semantic-based discovery protocol in an innovative m-commerce framework, and show its benefits.

Keywords: Ubiquitous computing; E-commerce; Bluetooth; resource discovery; Semantic Web; matchmaking.

1 Introduction

New mobile architectures allow for stable networked links from almost everywhere, and more and more people make use of information resources for leisure 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).

In their seminal work (Avancha et al., 2002) clearly pointed out the need and possibilities of an enhanced Bluetooth discovery protocol, although they did not provide neither a formal framework to obtain an approximate discovery service, nor proposed a fully-integrated semantic-enabled discovery protocol. Following their intuition, 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, adapting approaches that make the enhanced SDP both compatible with the existing SDP one –thus not troubling the standard– and able to smoothly integrate approaches and technologies developed for the Semantic Web.

The result of our work is an integrated "semantic layer" within the application level of the standard Bluetooth stack able to manage a simple interchange of semantically annotated information between a mobile client performing a query and a server exposing available resources.

In our framework 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 discover and return resources available in the m-marketplace. Both requests and resources are expressed as semantically annotated descriptions, so that a semantic similarity value can be computed as part of the ranking function, to choose most promising resources for a given request.

The remaining of this paper is structured as follows: next section revises basics of Bluetooth and of Semantic Web technologies we adapt to our setting. Then we move on to describe our semantic-enhanced SDP. An illustrative example in the m-commerce framework clarifies both motivations and behavior of our framework. Then we report on related work; future research and conclusions close the paper.

2 Basics

In this section, to make the paper self-contained, we briefly recall basics of technologies and languages our work is based on.

2.1 Bluetooth

Bluetooth (Bluetooth, 1999) is a wireless technology originally devised to replace the cable connection for portable and/or fixed devices also maintaining high security levels. It operates in the unlicensed 2.4-2.485 GHz frequency band also said ISM (Industrial, Scientific and Medical) and allows to reach data rates of up to 3 Mbps (supported for 2.0 + EDR standard version). The key features of Bluetooth technology are basically robustness, low power and low cost.

Bluetooth enabled handheld devices connect and communicate wirelessly building short-range ad hoc networks known as *piconets*. Each device can simultaneously maintain up to seven concurrent communications with other hosts within the same piconet, but each device can also belong to several piconets simultaneously. They are established dynamically as mobile nodes enter or leave the radio range.

An important Bluetooth feature is the capability to simultaneously handle both data and voice connections, hence, unlike many other wireless standards, Bluetooth specification gives product developers both link layer and application layer definitions, in order to support data and voice applications.

The operating range depends on environmental conditions but mostly on the device class. Possible classes are the third one (allowing a range of up to 1 meter), the second one -most commonly widespread in mobile devices- having a range of 10 meters and the first one class with a 100 meters radio range primarily used in industrial applications. The most common class 2 can consume up to 2.5 mW.

The Bluetooth stack specifications currently includes the baseband and the radio frequency layers, the Link Manager Protocol (LMP) for set-up, security and management of links among devices, a Host Controller Interface (HCI) which provides a command interface for baseband and Link Manager controllers, the Logical Link Control and Adaptation Protocol (L2CAP) able to implement multiplexing and logical connections toward higher protocol layers as well as segmentation and assemblage of large PDUs and finally the RFCOMM protocol (serial RS-232 port emulation).

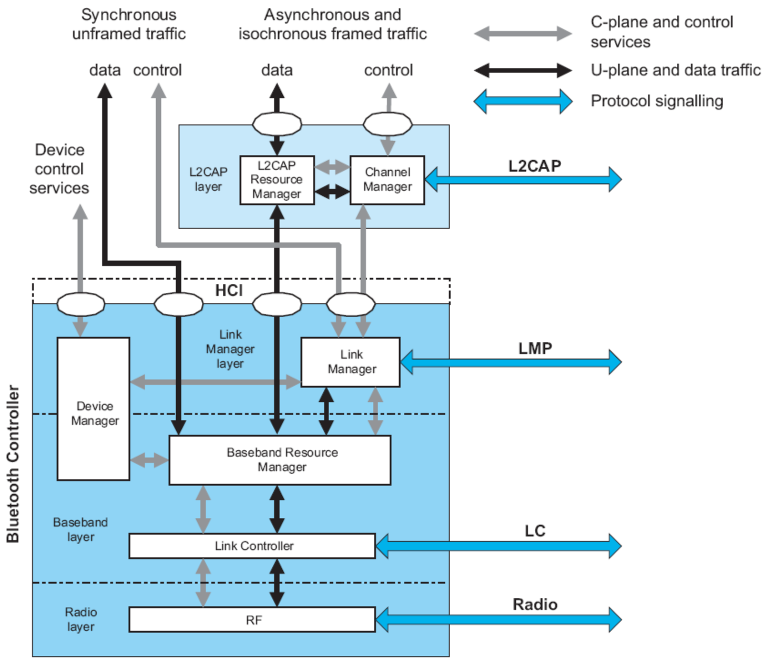


Figure 1 The Bluetooth core system architecture (Bluetooth, 1999)

The Service Discovery Protocol (SDP) is deployed over the previous stack levels — a service layer required by all Bluetooth applications. The discovery of resources in Bluetooth piconets takes place in a pattern-matching fashion: every service or attribute of that service is unambiguously labeled with a 128-bit Universally Unique IDentifier (UUID). Hence resource discovery is strictly syntactic and consequently SDP manages only exact matches.

This approach is well suited to discovery of handheld Bluetooth devices as, for example, a GPS antenna and a cellular phone. Each of those services is simple and straightforward enough, in terms of behavior and data exchange, to be associated with a numerical UUID. The phone will specify the GPS device UUID in its discovery request and then will wait for a positive or negative answer. In case of negative answers, it is not possible to discover services similar to the requested one.

2.2 Semantic Languages and Technologies

The "Semantic Web" initiative, launched in 2001 (Berners-Lee et al., 2001; Shadbolt et al., 2006), has as main objective to create an infrastructure where data can be seamlessly shared and reused across applications, adopting languages that allow the unambiguous interpretation of data meaning, that is providing an explicit semantics to those data. Although we are still far from fulfilling such an ambitious target, languages and technologies that have sprung from this initiative are already mature enough to be widely used and deployed in several applications.

The basic idea of the Semantic Web initiative is to annotate information by means of markup languages, based on XML, such as RDF, RDFS (RDF Primer-W3C Recommendation 10 February 2004,) and OWL (Ontology Web Language) (OWL, 2004). 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.

OWL comes in three sub-languages, namely:

- *OWL-Lite.* It allows class hierarchy and simple constraints on relation between classes.
- *OWL-DL.* Based on Description Logics (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.

We provide here a closer look at DLs, which are the formal logical foundations of our approach. DLs are a family of logic formalisms for Knowledge Representation (Baader et al., 2002; Borgida, 1995; Donini et al., 1996), also knows as Terminological or Concept languages, in a decidable subset of First Order Logic. In DLs, the basic syntax elements are:

- *concept* names, *e.g.*, $learningToy$, $geography$, $child$
- *role* names, like $stimulatesToLearn$
- *individuals*, like $Playstation$

Intuitively, concepts stand for sets of objects, and roles link objects in different concepts, *e.g.*, the role $stimulatesToLearn$ links learning toys to geography discipline. 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.*, $toy \sqcap \exists suitableFor.kid$, which indicates the set of toys whose users include kids) and *universal role quantification* (*e.g.*, $learningToy \sqcap \forall stimulatesToLearn.music$, which describes only musical toys). Other constructs may involve counting, as *number restrictions*: $teddyBear \sqcap (\leq\ 0\ hasPowerType)$ expresses peluches without any power supply, and $videoGame \sqcap (\geq\ 2\ allowedPlayers)$ describes game consoles allowing to play with at least two persons.

Many other constructs can be defined, up to create n-ary relations (Calvanese et al., 1998), 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 kids can be divided into males (boys) and females (girls) using the two inclusions: $kid \sqsubseteq boy \sqcup girl$ and $boy \sqsubseteq \neg girl$, or that a tale book can have only one subject as $book \sqsubseteq (\leq\ 1\ hasSubject)$.

Definitions are useful to give a meaningful name to particular combinations, as in $babyToy \equiv toy \sqcap \forall suitableFor.baby$. 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 $C \equiv D$ when $C^{\mathcal{I}} = D^{\mathcal{I}}$. A *model* of a TBox $\mathcal{T}$ is an interpretation satisfying all inclusions and definitions of $\mathcal{T}$. See Table 2 for details.

Adding new constructors makes DL languages more expressive. Nevertheless, this usually leads to a growth in computational complexity of inference services (Brachman and Levesque, 1984). Hence a trade-off is worthwhile.

In this work we refer to the *Attributive Language with unqualified Number restrictions* ($\mathcal{ALN}$) DL, a subset of OWL-DL. Constructs of $\mathcal{ALN}$ DL are the following ones (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.

- $(\geq\ n\ R)$[a], $(\leq\ n\ R)$, $(=\ n\ R)$[b], *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).

1. definition $microscope \equiv learningToy \sqcap \forall stimulatesToLearn.science$

2. inclusion $electricToy \sqsubseteq (\geq\ 1\ hasPowerType)$

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 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 DL 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, with “bushy TBoxes” (Di Noia et al., 2003).

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

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

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

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

3 Proposed Enhanced Bluetooth Service Discovery Protocol

3.1 Motivation and System Infrastructure

The basic Bluetooth SDP is surely inadequate when it comes to the need for handling complex requests. If we want to retrieve resources whose description cannot be classified within a rigid schema, a SDP more flexible than the standard one is needed. In our idea such a SDP should be able to carry out discovery of resources described using the rich formalism of semantic languages, and provide not only binary yes/no answers, but possibly an ordered list of available resources. This list, as in a classical Information Retrieval task must be sorted according to their –semantic– similarity to the request.

To motivate our objective, let us consider a mobile commerce context as reference scenario. A wireless hotspot can be used as central business directory containing shopping mall goods descriptions, whereas buyers may use their mobile devices in order to discover information about available services/resources.

Obviously the hotspot resources will be associated with a large number of complex descriptions. Hence, in that case, simple UUID structured requests, even if theoretically possible, would lead to a meaningless simplistic positive or negative answer. In order to address this issue, so increasing both the quality of resource discovery and the usefulness of Bluetooth SDP, we propose to use semantically annotated descriptions of services/resources rather than simple numeric identifiers, to be used in a semantic discovery process.

In the mobile environment adopted as case study scenario, a user connects via Bluetooth with the hotspot to submit her request in OWL/DIG formalism [c]. We assume the zone provider classifies resource contents by means of an OWL ontology, which have been previously collected as descriptions of mall goods/services. Each resource in the m-marketplace owns an URI and is semantically annotated by its OWL description. The hotspot must be endowed with a semantic discovery facilitator (in our system we adapt the *MAMAS-tng* reasoner (Di Noia et al., 2004)) measuring a "semantic similarity" value and providing as result a list of discovered resources matching the user request, ranked according to their degree of correspondence to the request itself.

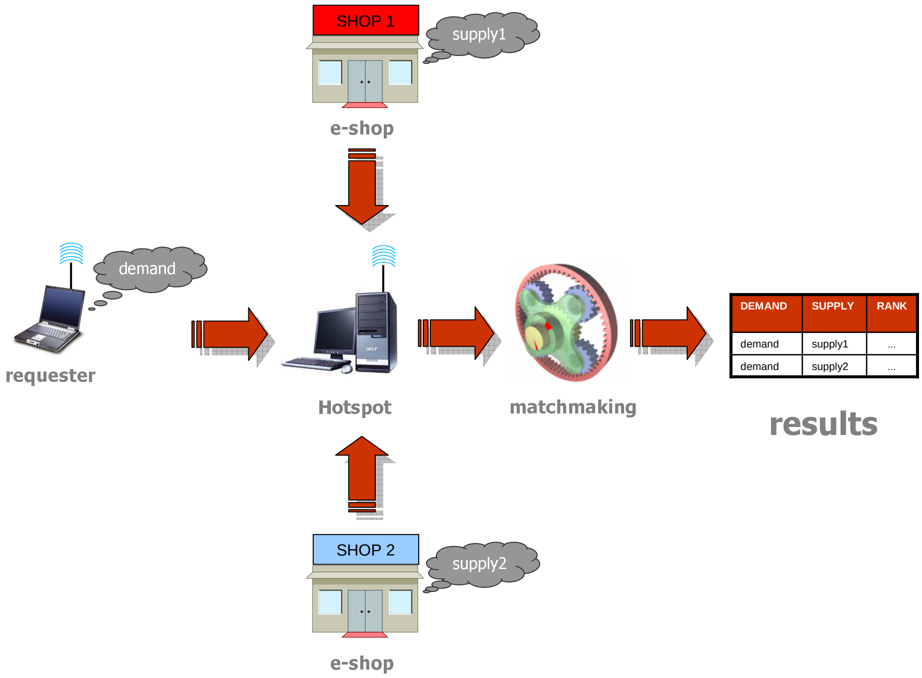


Figure 2 The proposed system infrastructure

3.2 The protocol

We aimed to implement a general resource discovery framework allowing co-existence of both syntactic and semantic discovery. By adding a semantic layer within the OSI Bluetooth stack at service discovery level, we maintained a backward compatibility (handheld device connectivity) also enriching the standard with the support to discovery of semantically annotated resources.

In order to illustrate the proposed enhancements to the basic Bluetooth SDP structure, we refer to a generic interaction between a requester and a resource provider, showing involved packet frames exchanged at the application layer.

[c]Obviously, we do not expect users to exploit any logic formalism. We expect users be provided with a friendly graphical application, see *e.g.*, the one we developed for our prototype framework, which is described later on

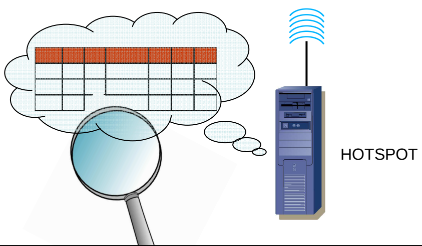
The requester–hotspot interaction starts after the user joins the MANET. She is able to ask for a specific service/resource by submitting a (semantic based) request. Recall that Bluetooth SDP is based on a request/response packet transfer occurring between SDP-layer client and the corresponding server (Bluetooth, 1999). With respect to the pattern-based UUID discovery, the semantic approach requires a further stage useful to select resource descriptions suitable for the semantic discovery process.

We associated reserved classes of 128 bit UUIDs in the original Bluetooth to label each specific ontology so calling this identifier *OUUID* (Ontology Universally Unique IDentifier). In such a way, a preliminary selection of resource descriptions that do not refer to the same ontology of the request (Chakraborty et al., 2001) can be performed by means of a simple OUUID based discovery procedure. Such preliminary stage does not identify a single service, but directly the context of resources we are looking for, that is a class of similar services.

It is important to remark that basically both semantic and syntactic discovery are based on the same original SDP framework. In the semantic based approach no modifications are made to the original structure of transactions, but simply we differently use the Bluetooth SDP procedures.

A user searching for a resource must initially communicate to the hotspot one or more ontology identifiers ($OUUID_R$) she manages. The hotspot selects OUUIDs matching each received $OUUID_R$ and replies to the client. Hence the requester composes a request (R) referred to the agreed ontology and sends it to the provider. The hotspot extracts descriptions of resources –cached within the hotspot itself– classified with the previous $OUUID_R$; furthermore it performs the semantic discovery process between R and selected resources it shares. Taking into account the discovery results, all the resources are ranked with respect to R and the sorted list is returned to the user.

The hotspot manages a registry containing all the semantically annotated resources belonging to the m-marketplace. A 32-bit identifier (called *SemanticResourceRecordHandle*) is uniquely associated to each resource record within the registry.



OUUID	Resource Name	Resource Description	Resource Utility Attr1	Resource Utility Attr2	Resource Attr6	…	Resource AttrN

Figure 3 The resource registry structure within the hotspot

Each record contains general information about a single semantic based resource and it entirely consists of a list of resource attributes, see Figure3. In addition to the OUUID one –described above– we have the following:

- *ResourceName*: a text string containing a human-readable name for the resource
- *ResourceDescription*: the DIG resource description (text string)
- *ResourceUtilityAttr_i*: numeric values used according to specific applications –*contextual parameters*

Contextual parameters have been introduced to be associated to environmental attributes of a resource (Lee and Helal, 2003); 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 itself).

Summarizing, the discovery procedure takes place in a two step mode. The first one is syntactic-based and aims to select resource descriptions related with the request via OUUID, whereas the second one is semantic-based and aims to select only the best available resources. From now on we will indicate the first discovery phase as *resource class discovery* and the second one as *resource content discovery*. In what follows, for each of them, we outline the structure of the SDP PDUs we added within the original framework to enable the semantic resource discovery.

Resource Class Discovery The *resource class discovery* is basically similar to the original Bluetooth SD. It is based on a request/response interaction making use of two PDUs: the ***SDP_OntologySearchRequest*** and the the ***SDP_OntologySearchResponse***.

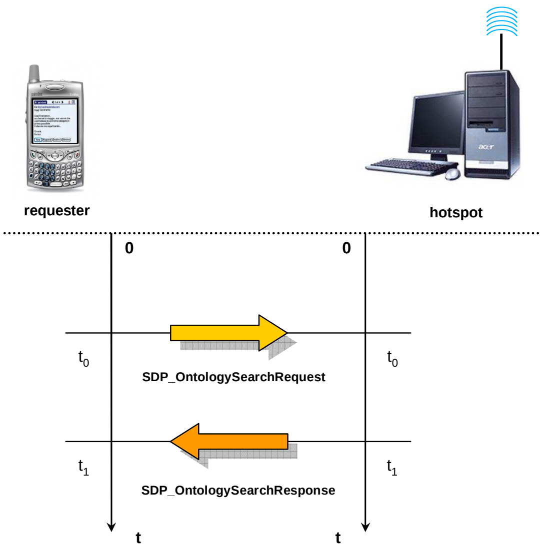


Figure 4 Resource class discovery

Considering the basic structure of a Bluetooth SDP frame sketched in the Figure5, we fill the parameters payload in a different way according to different used packets.

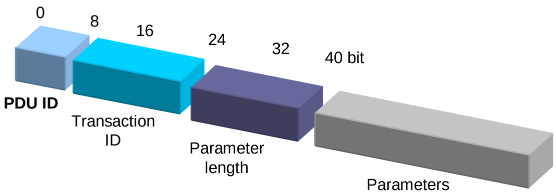


Figure 5 General Service Discovery Protocol PDU structure

The ***SDP_OntologySearchRequest*** PDU is depicted in Figure6. There are two parameters. The *OntologySearchPattern* is a variable data element sequence whose single component is a OUUID. The sequence must contain at least 1 and at most 12 OUUIDs, as in the original standard. The *ContinuationState* parameter maintains the same purpose of the original Bluetooth (Bluetooth, 1999), that is if the SDP request needs more than a single PDU, the SDP server generates a partial response and the SDP client waits for next part of the complete answer. Its length can range from 1 to 17 bytes.

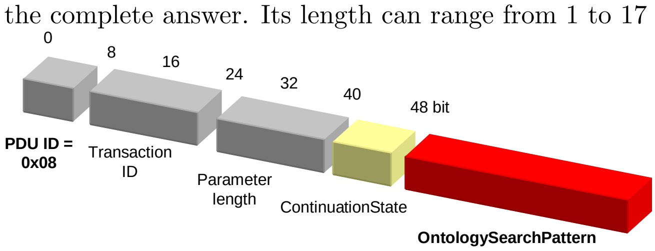


Figure 6 *SDP_OntologySearchRequest* parameters

The ***SDP_OntologySearchResponse*** PDU is generated by the hotspot as a reply to the previous frame. Figure7 shows its parameters. The *TotalOntologyCount* is a 16-bit integer containing the number of ontology identifiers matching the requested ontology pattern. Whereas the *OntologyRetrievedPattern* is a variable data element sequence where each element in the sequence is a OUUID, matching at least one among those 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 later on the ontology browsing mechanism for further details). Hence the pattern sequence contains at least 1 and at most 12 OUUIDs.

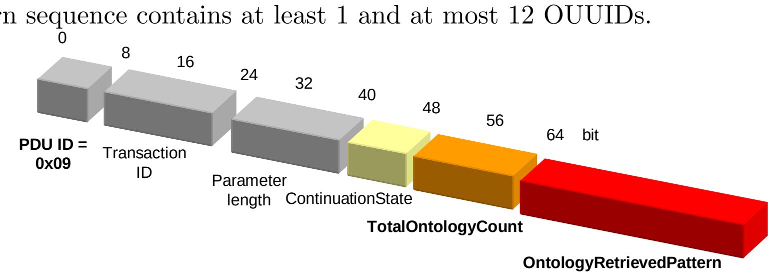


Figure 7 *SDP_OntologySearchResponse* parameters

Resource Content Discovery The *resource content discovery* follows the preliminary discovery phase described above. It provides, together with the original SDP capabilities, further semantic enabled features. Also in this case we have a request/response interaction exploiting two PDUs: the ***SDP_SemanticServiceSearchRequest*** and the ***SDP_SemanticServiceSearchResponse***

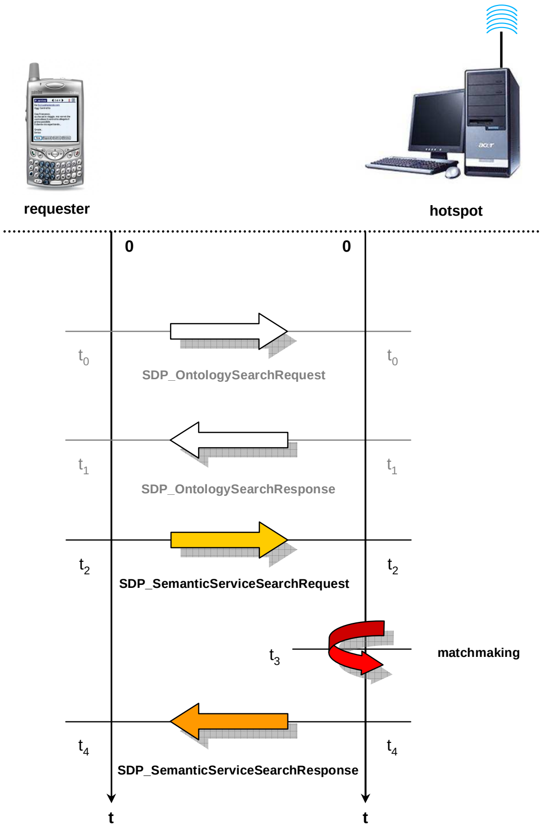


Figure 8 Resource content discovery

The ***SDP_SemanticServiceSearchRequest*** PDU is pictured in Figure9. The parameters field contains the *SemanticResourceDescription* (*i.e.*, a variable data element text string in DIG formalism representing the resource we are searching for), the *ContextAwareParam1* and *ContextAwareParam2* (data element unsigned integers). As an example, 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 time of the scheduled departure of the flight. The other parameters maintain the original Bluetooth usage (Bluetooth, 1999).

The ***SDP_SemanticServiceSearchResponse*** PDU is the server reply to the previous PDU. In Figure10 the frame structure is sketched, with related 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.

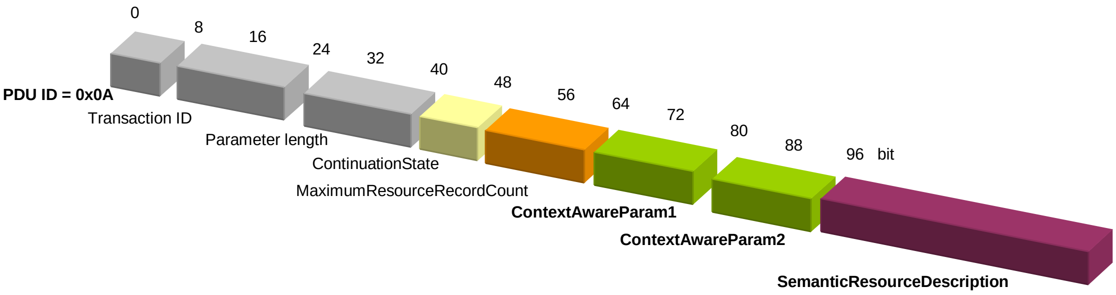


Figure 9 *SDP_SemanticServiceSearchRequest* parameters

Its length is given by $4 \cdot CurrentServiceRecordCount$ bytes. The other parameters maintain the same purpose of the original Bluetooth (Bluetooth, 1999).

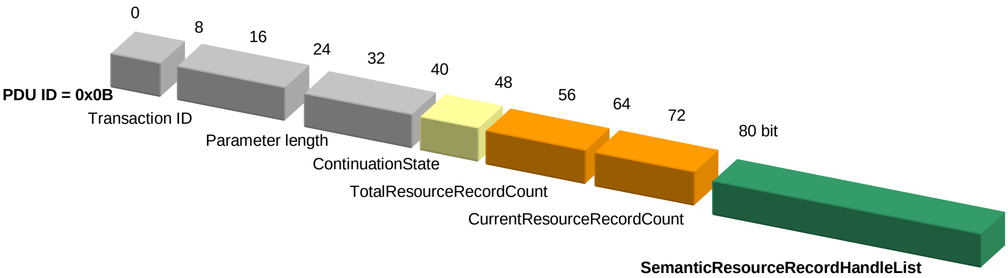


Figure 10 *SDP_SemanticServiceSearchResponse* parameters

Notice that in all previous cases, the error handling is managed with the same mechanisms and techniques of Bluetooth standard (Bluetooth, 1999).

In oder 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 (Bluetooth, 1999). Hence resulting types are as it is shown in Table4.

Finally Table5 shows the overall PDU types in the enhanced version of the Bluetooth Service Discovery Protocol.

3.3 Ontology management

Each resource retrieval session starts after setting between client and server a shared pattern of ontology identifiers (OUUIDs). It is noteworthy that, 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 (*i.e.*, what OUUIDs) are handled by the zone server also without having any a priori information about resources. For this purpose we use the service browsing feature (Bluetooth, 1999) in a slightly different fashion with respect to 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*, which contains a list of OUUIDs. Each of them is the *browse group* a resource may be associated with for browsing.

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

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 and related descriptions

Groups are organized in a hierarchical fashion, hence when a client wants to browse a hotspot resource classes, she can create an ontology search pattern containing the OUUID that represents the entry point for classes browsing.

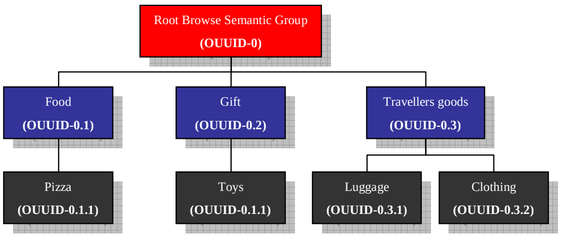


Figure 11 Ontology browsing hierarchy

Generally a hotspot will support 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.

Having determined the resource category and the corresponding reference ontology, the client can also download it from the hotspot.

Notice that, since the proposed approach is fully compliant with Semantic Web technologies, the user may exploit the same semantic enabled descriptions she may use in other Semantic Web compliant frameworks (for example, the web site of a shopping mall). That is, there is no need for different customized resource descriptions and modeling, if the user employs different applications either on the web or on a mobile system. The formal semantics of the descriptions is unique with respect to the reference ontology and can be shared among different environments.

3.4 Additional features

In e-commerce scenarios, the match between request and good 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 should depend not only on the similarity between the (semantic-enabled) description of the demand and of the supply, but also on price, quantity, delivery time differences, *i.e.*, data-type properties. An overall utility function has to combine all features to return a significative global match degree.

Also notice that, in m-commerce applications, in addition to the above mentioned parameters also other *context-aware* variables may influence discovery 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 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, and explains the motivation for introducing data-type attributes in the SDP PDUs.

As an example, the utility function we used in our framework 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) and *s_match* (score computed during the semantic discovery process through *rankPotential* algorithm (Di Noia et al., 2004))

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

Notice that p_D is weighted by a $(1 + \alpha)$ factor. The idea behind this weight is that, usually, the requester may be 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.

3.5 Evaluating the approach

In evaluating the semantic-enhanced resource discovery performances, care has to be paid in avoiding a too strict comparison with the basic Bluetooth service discovery. In fact the two approaches are slightly different from the point of view of the protocol architecture but deeply different for the kind of resources handled and returned results.

Basically we have two fundamental differences between a standard service discovery Bluetooth architecture and the enhanced one. The latter presents a higher complexity of the framework as well as longer response times. The complexity is repaid by the quality of resource retrieval.

In the proposed framework, additional overhead with respect to the original Bluetooth SDP is introduced only by *resource content discovery* phase (recall that previous *resource class discovery* is basically analogous to the UUID-based approach). Hence a correct performance comparison must take into account increased response time produced by the additional interaction step.

Generally, a simple request/response interchange PDU is enough to carry out the semantic-based discovery phase on the whole. In some cases (*i.e.*, in case of several resources discovered with associated images) we may have two or more PDUs as reply. Nevertheless it would be incorrect to affirm this interaction is comparable (in terms of response time) to a standard SDP PDUs interchange because, request messages in the enhanced SDP are 10 to 30 times larger than corresponding requests in the regular standard as well as reply frames generally are up to 100 times larger than Bluetooth SDP response ones.

However, in absolute terms, the most significant piece of response time is the computation time contributed by the semantic facilitator, that is the time spent by the hotspot in the discovery procedure. Generally speaking, the increased response time is acceptable from the user perspective because it ranges between 1 and a few seconds. It is obvious that in the enhanced approach response time increases, whereas it is nearly constant in case of standard Bluetooth SDP where parsing and matching mechanisms are trivial.

Hence the penalty paid for using a complex semantic-based discovery framework has to be considered in an evaluation of the system, but basically it seems to be acceptable from a user considering obtained added-value on the results.

4 Illustrative Example

We present here a simple running example to illustrate our framework, together with *MicroClerk*, our prototype application designed to exploit the enhanced Bluetooth SDP. The case study is analogous to the one presented in (Avancha et al., 2002), and we face it by means of our approach.

Let us suppose a user is in a airport, where a semantic-enhanced Bluetooth hotspot is available. She is waiting for her flight to come back home and she is endowed of a PDA equipped with *MicroClerk*. 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, but she has not much time to go before her flight departure. 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 such a request is too complex to be expressed by means of standard UUID Bluetooth SDP mechanism (actually, the Bluetooth SDP was not even thought for such a kind of applications). In addition, non-exact matches between resource request and offered ones are highly probable and the on/off matching system provided by the original standard in this case could be largely inefficient.

In what follows a typical interaction client/hotspot is hence presented. First of all, *MicroClerk* allows the composition of the OUUID pattern to submit to the server. The hotspot will reply with supported ontologies and the user chooses one of them. Hence the ontology is parsed by *MicroClerk* in order to allow the ontology navigation (see Figure12).

The user may now compose her request via the intensional navigation of the shared ontology (see Figure13). The composition of the user request takes place in an interactive fashion. The user scrolls the ontology in a vertical mode selecting root reference classes, but also in a horizontal mode going down in the class hierarchy. With respect to roles, when a user selects an attribute *MicroClerk* will present compatible fillers. Finally a recap screen is displayed whenever a selection is performed, as pictured in Figure14.

When the request is ready to be submitted, the user sends it to the hotspot. The semantic resource request is expressed as a DIG statement exploiting DL semantics and encapsulated in the SDP PDU, as described in the previous Section.

The semantic-enabled hotspot collects the request and initially selects resources expressed by means of the same ontology shared with the requester. Hence a primary selection of suitable resources is performed. In addition, the semantic facilitator carries out the discovery process between each offered resource in the m-marketplace and the requested one computing a "semantic similarity" (*s_match*) (Colucci et al., 2005). Finally the results are ranked and returned to the user, as it is shown in Figure15. Results are presented as a list ranked according to semantic similarity criteria but also taking into account the compliance of resource context-

Figure 12 OUUID selection snapshot

Figure 13 Request composition graphical user interface

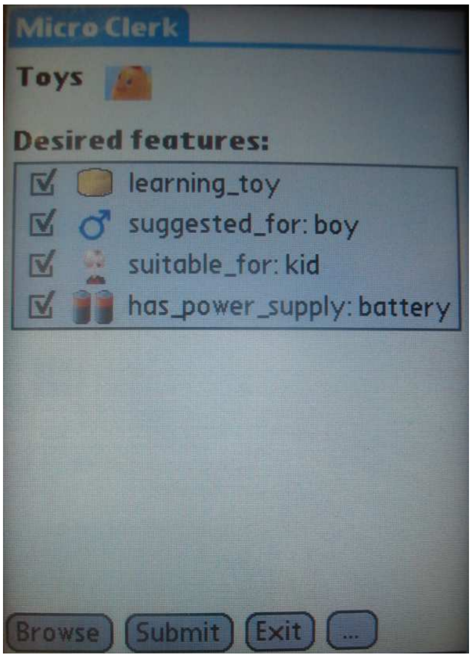


Figure 14 Request recap screen

aware parameters with respect to the ones provided with the request. Selecting one of provided results more details (such as contextual parameter values) are displayed together with (if available) a sample image of the good (see, *e.g.*, Figure16).

To better clarify the discovery process a subset of the ontology used as a reference in the example is reported in Figure17. For the sake of simplicity, only the class hierarchy and disjoint relations are presented. Let us suppose the hotspot Knowledge Base is populated with the following resources 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 $\boldsymbol{D}$ 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 the user sets a reference price of 200$ ($p_D = 200$) as well as the scheduled departure time within 30 minutes ($t_D = 30$).

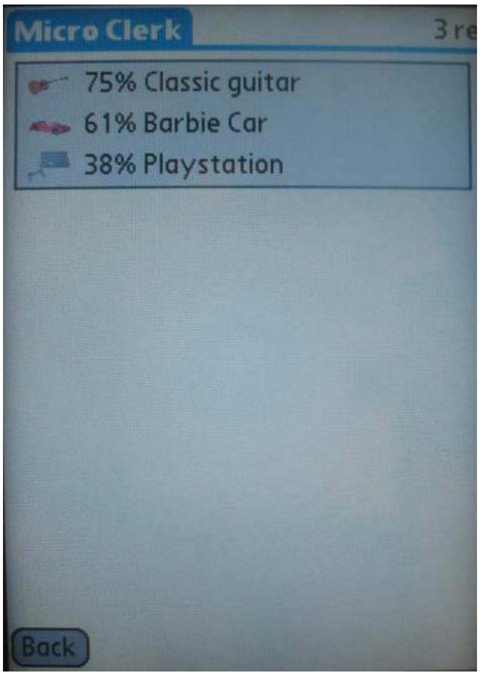


Figure 15 Semantic Discovery results

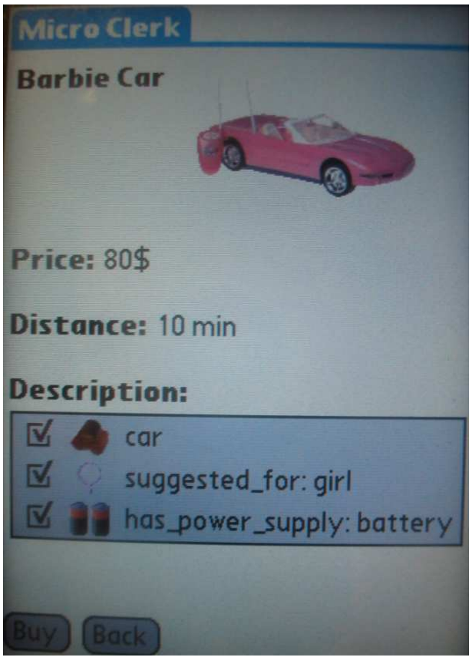


Figure 16 Result details snapshot

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

Figure 17 The ontology used in the example

In Table6 retrieval results are presented. The second column shows whether each retrieved resource is compatible or not with request D and, in case, the computed semantic score. In the fourth column, such results are expressed in a relative form between 0 and 1 to allow a more immediate comparison among requests and different resources, and to put in a direct correspondence various rank values. Finally, in the last column results for the overall discovery results are shown.

request/resource	compatible	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 6 Discovery results

Notice that using only semantic match values (*s_match*), *Playstation* is the second best choice for the demander and *Barbie_car* the third one. On the other hand, taking into account context-aware information related to price and physical distance, the order is reversed. 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.

After having selected the best resource, the server of the chosen virtual shop may 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 easily achievable exploiting the above SDP infrastructure.

5 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 (Avancha et al., 2002). 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 discovered services. Neither was formally modeled a complete Semantic Discovery Protocol, which is needed to obtain a fully integrated framework within Bluetooth.

In recent years dynamic distributed systems have been developed adopting various technologies and for different purposes. In (Chen et al., 2001) a Jini-based distributed agent framework was used in a hybrid agent-oriented/service-oriented approach, whereas in (von Hessling et al., 2004) 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 (Chen et al., 2001) and in (Chakraborty et al., 2001), is due to the lack of shared ontology support. In (Chakraborty et al., 2001) 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/resources only to those semantically suitable.

In (von Hessling et al., 2004) a mobile environment was 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 provided.

In (Sundramoorthy et al., 2003) SDP@HA was 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 the 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 (Sundramoorthy et al., 2003) 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 (Chen et al., 2001) presented 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 approximate 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 (Chen et al., 2001) there is no mention to the solution of the approximate 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 was investigated in (Prestes et al., 2004). There an example of collaborative environment was 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.

(Liu et al., 2002) introduced 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 (Lee and Helal, 2003), 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.

In (Ruta et al., 2005) a collaborative environment for semantic enabled mobile devices in peer to peer scenarios was proposed, exploiting an initial version of our approach.

Significant applications of semantic ad-hoc networks can be made in all fields 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, for example the *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. (Watson et al., 2004) 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.

6 Conclusion and Future Work

We have presented and motivated a semantic-enhanced Bluetooth discovery protocol in an m-commerce framework.

This work has, in our opinion, manifold contributions: it provides an enhanced semantic-based discovery protocol, augmenting Bluetooth with a whole set of new possible applications in m-commerce; it keeps the enhanced discovery protocol compatible with the original one, thus permitting the coexistence of both protocols; it paves the way to bridging the gap between Bluetooth applications and emerging Semantic Web technologies and languages.

In the near future the mobile platform will also provide a negotiation support allowing to modify some request features –adding, removing, setting to strict– in order to find a resource best fitting the user request in a progressive refinement fashion.

Acknowledgements

The authors are grateful to Floriano Scioscia for useful implementations. They would also like to acknowledge the partial support of FAR project PITAGORA (Piattaforma Telematica per la Informazione e la Gestione dei Sistemi di Trasporto Collettivo) and of Apulia Region Strategic Projects: "Telecommunication Facilities and Wireless Sensor Networks in Emergency Management" and "Distributed Production as Innovative System".

References and Notes

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

Baader et al., 2002 Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., and Patel-Schneider, P. (2002). *The Description Logic Handbook*. Cambridge University Press.

Berners-Lee et al., 2001 Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 248(4):34–43.

Bluetooth, 1999 Bluetooth. http://www.bluetooth.com.

Borgida, 1995 Borgida, A. Description Logics in Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):671–682.

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

Calvanese et al., 1998 Calvanese, D., De Giacomo, G., and Lenzerini, M. (1998). 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.

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

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

Colucci et al., 2005 Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Mongiello, M. (2005). Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. *Electronic Commerce Research and Applications*, 4(4):345–361.

Di Noia et al., 2003 Di Noia, T., Di Sciascio, E., Donini, F., and Mongiello, M. (2003). Abductive matchmaking using description logics. In *IJCAI '03*, pages 337–342.

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

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

Gryazin, 3pdf Gryazin, E. (http://www.hpl.hp.com/techreports/2002/HPL-2002-233.pdf). Service Discovery in Bluetooth.

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

Liu et al., 2002 Liu, J., Zhang, Q., Li, B., Zhu, W., and Zhang, J. (2002). 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.

OWL, 2004 OWL (2004). www.w3.org/TR/owl-features/.

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

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

Ruta et al., 2005 Ruta, M., Di Noia, T., E.Di Sciascio, Donini, F., and Piscitelli, G. (2005). Semantic based collaborative p2p in ubiquitous computing. In *IEEE/WIC/ACM WI-2005 Web Intelligence*, pages 143–149.

Shadbolt et al., 2006 Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

Sundramoorthy et al., 2003 Sundramoorthy, V., Scholten, H., Jansen, P., and Hartel, P. (2003). Service Discovery at Home. In *4th IEEE Pacific-Rim Conf. On Multimedia.*

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

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