Semantic Matchmaking for Location-Aware Ubiquitous Resource Discovery

Michele Ruta, Floriano Scioscia, Eugenio Di Sciascio, Giacomo Piscitelli
Politecnico di Bari
Via Re David 200
I-70125, Bari, Italy
{m.ruta, f.scioscia, disciascio, piscitel}@poliba.it

***Abstract*— Semantic technologies can increase effectiveness of resource discovery in mobile environments. Nevertheless, a full exploitation is currently braked by limitations in stability of data links and in availability of computation/memory capabilities of involved devices. This paper presents a platform-independent mobile semantic discovery framework as well as a working prototypical implementation, enabling advanced knowledge-based services taking into account user's location. The approach allows to rank discovered resources based on a combination of their semantic similarity with respect to the user request and their geographical distance from the user itself, also providing a logic-based explanation of outcomes. A distinguishing feature is that the presented mobile decision support tool can be proficiently exploited by a nontechnical user thanks to careful selection of features, GUI design and optimized implementation. The proposed approach is clarified and motivated in a ubiquitous tourism case study. Performance evaluations are presented to prove its feasibility and usefulness.**

Keywords-Ubiquitous Computing; Semantic Web; Resource Discovery; Matchmaking; Location-based Services; Human-Computer Interaction

I. INTRODUCTION

Mobile solutions for semantic-based geographical resource discovery [1] are a growing research and business opportunity, as a growing number of people make use of informative resources exploiting mobile systems [2]. The ICT (Information and Communication Technology) paradigm "anytime and anywhere for anyone" is nowadays deeply actual, but some practical aspects hinder a widespread diffusion of concrete and useful advanced applications. In ubiquitous computing scenarios, information technology can assist users in discovering resources, thus aiding people to retrieve information satisfying their needs and/or giving more elements to make rational decisions. Nevertheless, when stable network infrastructures are lacking and exploited devices are resource-constrained, the process of supporting the user searching goods or services is a challenging subject [3].

Techniques and ideas of the Semantic Web initiative are potential means to give flexibility to discovery [4]. In fact, Semantic Web technologies applied to resource retrieval open new possibilities, including: (i) formalization of annotated descriptions that become machine understandable so enabling interoperability; (ii) reasoning on descriptions to infer new knowledge; (iii) validity of the Open World Assumption (OWA) (what is not specified has not to be interpreted as a constraint of absence) [5], overcoming limits of structured data models. Though interesting results have been obtained in the evolution of canonical service discovery in the Web, several issues are still present in ad-hoc and ubiquitous environments, because of both host mobility and limited capabilities of mobile devices. Hence, many people equipped with handheld devices usually prefer traditional fixed discovery channels so renouncing to an instant fruition of resources or services. Nevertheless, the rising potentialities of wireless-enabled handheld devices today open new possibilities for implementing flexible discovery approaches.

This paper proposes a general framework which enables a semantic-based location-aware discovery in ubiquitous environments. It has been implemented in a mobile Decision Support System (DSS) whose main goal is to allow users equipped with handheld devices to take advantage of semantic resource annotation and matchmaking as well as of logic-based ranking and explanation services, while hiding all technicalities from them and letting to interact with the system without requiring dependable wired infrastructures.

In order to better clarify the proposed settings and the rationale behind them, a u-tourism [6] case study is presented. The proposed approach allows to perform a selective resource discovery based on proximity criteria. Since users equipped with PDAs or smartphones are dipped in a pervasive environment, they could be specifically interested in resources or services near them. Hence, during discovery, resources/services close to the user should be ranked better than the ones far off (supposing an equivalent semantic distance from the request). In other words, the semantic distance between request and an offered resource should be properly rectified taking into account the physical distance occurring between user and resource itself (supposing it has an environmental collocation). In the proposed touristic virtual guide application, this feature has been implemented by means of the integration of a positioning module within the discovery tool. The application recognizes the user location and grades matchmaking outcomes according to vicinity criteria.

The retrieval process is accomplished across multiple steps. Request formulation is the most important one. It is particularly critical in case of ontology-based systems: the query language has to be simple but, in the same way, its

expressiveness must allow to correctly express user requirements. A selective retrieval of what the user is really looking for has to be so enabled. The paper faces the above issues and presents a general framework which allows semantic-based matchmaking and retrieval, exploiting an intuitive Graphical User Interface (GUI).

Main features of the proposed approach are:

- Full exploitation of non-standard inferences introduced in [7] to enable explanation services and bonuses calculation;
- Semantic-based ranking of retrieved resources;
- Fully graphical and usable interface with no prior knowledge of any logic principles;
- No physical space-temporal bonds in system exploitation.

The proposed tool can be considered as a subsidiary system to be exploited whenever more comfortable means to perform a resource discovery are not available. It is a general-purpose mobile DSS where the knowledge domain is encapsulated within a specific ontology the user must select at the beginning of her interaction with the system. The knowledge about a domain can be exploited, in order to derive new information from the one stated within metadata associated to each resource.

Since the interest domain is modeled with an OWL (Web Ontology Language) [8] ontology, the user is able to browse the related knowledge starting from "her vague idea" about the resource she wants to discover. By means of a preliminary selection of the reference ontology, the user focuses on a specified scenario, so determining the context for the following interactions with the system. Different sessions in the application exploitation could refer to different ontologies and then could entail interactions with the system aimed at different purposes. For example a generic user could exploit the application as a pocket virtual guide for tourist purposes selecting a cultural heritage ontology and in a further phase, after concluding her visit, she can adopt it as a mobile shopping assistant to buy goods in a B2C m-marketplace: in that case, she will select an e-commerce ontology. Once the request has been composed, its formal relations are exploited, in order to find resources able to satisfy user requirements. Based on the formal semantics of both the request and the returned resource/service descriptions, an explanation of the matchmaking outcome is then provided to foster further interaction.

The remaining of the paper is structured as follows: in the next section, motivation of the paper is outlined and in the third section basics of matchmaking and exploited Description Logics inference services are briefly recalled. Sections IV and V describe the framework and the implemented prototypical system, respectively. The subsequent section helps to understand and justify the approach through a case study referred to a u-tourism scenario. Some performance evaluation is reported in Section VII, while Section VIII discusses related work. Finally, conclusion and future research directions close the paper.

II. MOTIVATION

Service/resource discovery is a challenging task. Finding resources and/or services encountering user needs often requires too much effort and time, especially when a user has just a vague idea of what she wants.

Several issues concerning traditional service discovery are exasperated in "evanescent" scenarios such as ubiquitous environments, due to both host mobility and limited capabilities of mobile devices. Small displays, uncomfortable input methods, reduced memory availability and low computational power restrain the exploitation of such applications. Hence usually, many people equipped with handheld devices still tend to prefer traditional fixed discovery channels (*e.g.*, via a PC), so renouncing to an instant fruition of resources or services.

Nevertheless the rising potentialities of wireless-enabled handheld devices provide the needed basic requirements for implementing flexible discovery frameworks. They involve advanced techniques permitting to find and share information more easily and more effectively. The final aim is to reduce the human effort in resource retrieval procedure, also granting an acceptable level of accuracy and coping with user mobility and heterogeneous scenarios. In most cases users are unable to exploit logic formulas needed to use a formal ontology; they want a simple visual representation to manipulate the domain of interest. A suitable discovery framework should be able to rapidly retrieve resources according to beneficiary's interests and to present them in an appealing fashion that facilitates examination and checking of their features.

Techniques and ideas of the Semantic Web initiative [9] are potential means to give flexibility to discovery [4]. In fact, Semantic Web technologies applied to resource retrieval open new possibilities, including:

- Formalization of annotated descriptions that become machine understandable so enabling interoperability;
- Reasoning on descriptions and inference of new knowledge;
- Validity of the Open World Assumption (OWA) (what is not specified has not to be interpreted as a constraint of absence) [5], overcoming limits of structured data models.

From this standpoint, the possibility of going beyond physical boundaries of structured and fixed network infrastructures is a significant added value. That is, a concrete exploitation of semantics in mobile contexts could enable further applications improving the trust of users in service fruition "from everywhere" [3].

III. BACKGROUND

A. Matchmaking Basics

Given *R* (for Request) and *O* (for Offer) both consistent with respect to an ontology $\mathcal{T}$, logic-based approaches to matchmaking proposed in the literature [10,11] use classification and consistency to grade match results in five categories:

- *Exact.* All the features requested in *R* are exactly the same provided by *O* and vice versa – in formulae $\mathcal{T} \models R \Leftrightarrow O$.
- *Full-Subsumption.* All the features requested in *R* are contained in *O* – in formulae $\mathcal{T} \models O \Rightarrow R$.
- *Plug-In.* All the features offered in *O* are contained in *R* – in formulae $\mathcal{T} \models R \Rightarrow O$.
- *Potential-Intersection.* There is an intersection between features offered in *O* and the ones requested in *R* – in formulae $\mathcal{T} \not\models \neg (R \sqcap O)$.
- *Partial-Disjoint.* Some features requested in R are conflicting with some other ones offered in *O* – in formulae $\mathcal{T} \models \neg (R \sqcap O)$.

A toy example will clarify differences among previous match types; let us suppose a tourist is making a visit and she is interested in seeing "medieval palaces with courtyards" (this is what we the previously named *R*). If there is a resource $\mathbf{O}_{\mathbf{exact}}$ annotated as "medieval palace with a courtyard", *R* and *O* coincide. From a matchmaking perspective, **Exact** matches are obviously the best, because both *R* and *O* express the same preferences and, since all the resource characteristics requested in *R* are semantically implied by *O* (and vice versa), the user finds exactly what she is looking for. On the contrary, if there is $\mathbf{O}_{\mathbf{full}}$ annotated as "medieval palace with a courtyard and frescoed roofs", all requirements in *R* are satisfied by *O*, but other non-conflicting characteristics are also specified in the returned resource. In a **Full** match, all the interpretations for *O* are surely also interpretations for *R* and then *O* completely satisfies *R*. This means that all the resource characteristics requested in *R* are semantically implied by *O* but not, in general, vice versa. Then, in a full match, *O* may expose some unrequested characteristics. From a requester's standpoint, this is not a bad circumstance, since anyway characteristics she was looking for are satisfied. If the provided resource is $\mathbf{O}_{\mathbf{plug-in}}$ simply labeled as "medieval palace", all characteristics in *O* were required by *R*, but the requirement of a courtyard is not explicitly satisfied. **Plug-In** match expresses the circumstance when *O* is more generic than *R*, and then it is possible that the latter can be satisfied by the former. Some characteristics in *R* are not specified, implicitly or explicitly, in *O*. This is surely more appealing for the provider than for the requester (as said, here we adopt the OWA). In case the returned resource is $\mathbf{O}_{\mathbf{potential}}$ annotated as a "medieval palace with a frescoed roof", neither all elements in *R* are in *O* nor vice versa. *R* and *O* are still compatible, since an explicit conflict does not occur. With **Potential** match it can only be said that there is some similarity between *O* and *R*, hence *O* might potentially satisfy *R*; probably some features of *O* are underspecified in its description, so the requester should contact the provider to know something more about them. Finally, supposing $\mathbf{O}_{\mathbf{partial}}$ is a "medieval church with a courtyard", a requirement in *R* is explicitly violated by *O*, making the provided resource incompatible with the request. **Partial** match states that *R* and *O* are conflicting (as evident a church cannot be represented as a palace), yet notice that the disjointness between them might be due only to some – maybe negligible from the requester's standpoint – incompatible characteristics. Hence, after a revision of opposed features, an agreement can be reached.

Standard logic-based matchmaking approaches usually allow only a categorization within match types. But while exact and full matches can be rare, a user may get several potential and partial matches. Then, a useful logic-based matchmaker should provide an ordering of available resources with respect to the request, but what one would get using classification and consistency is a Boolean answer. Also partial matches might be just "near miss", *e.g.*, maybe just one requirement is in conflict, but a pure consistency check returns a hopeless false result, while it could be interesting to order "not so bad" ads according to their similarity to the request.

B. Description Logics Inference Services

The proposed approach is grounded on Description Logics (DLs), a family of logic formalisms for Knowledge Representation, also known as Terminological languages, in a decidable fragment of First Order Logic [5].

Basic syntax elements are: *concept* names, *role* names, and *individuals*. They can be combined using *constructors* to build concept and role *expressions*. Each DL exposes a different set of constructors. A constructor used in every DL is the *conjunction* of concepts, usually denoted as $\sqcap$; some DLs include also disjunction $\sqcup$ and complement $\neg$ (to close concept expressions under Boolean operations). Roles can be combined with concepts using *existential role quantification* and *universal role quantification*. Other constructs may involve counting, such as *number restrictions*. Many other constructs can be defined, so increasing the expressiveness of the language. Nevertheless, this usually leads to a growth in computational complexity of inference services [12]. Hence a trade-off is worthwhile.

OWL DL [8] is a W3C (World Wide Web Consortium) standard language for the Semantic Web, based on DLs theoretical studies. It allows a satisfactory expressiveness keeping computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) in reasoning procedures. OWL DL includes all OWL language constructs with restrictions such as type separation (a class cannot also be an individual or property, a property cannot also be an individual or a class) maintaining interesting computational properties for a concrete application of reasoning systems in various common scenarios.

In this paper, we refer to the $\mathcal{ALN}$ (Attributive Language with Unqualified Number Restrictions) subset of OWL DL, which has polynomial computational complexity for standard and nonstandard inferences. Constructs of $\mathcal{ALN}$ DL are reported hereafter (see Table I 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*.
- $\neg$, *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*.
- $\geq n\ R$, $\leq n\ R$, $=n\ R$, *unqualified number restrictions*. Respectively the minimum, the maximum and the exact number of objects participating in the relation *R*. Notice that $\exists$R is semantically equivalent to ($\geq$1R) and that (=nR) is a syntactic shortcut for ($\geq$n R) $\sqcap$ ($\leq$ nR).

- TABLE I. SYNTAX AND SEMANTICS OF $\mathcal{ALN}$ DL CONSTRUCTS

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\varnothing$
intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
atomic negation	$\neg A$	$\Delta^{\mathcal{I}} - A^{\mathcal{I}}$
universal quantification	$\forall R.C$	$\{d_1 \in \Delta \mid \forall d_2 \in \Delta: (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$
concept inclusion	$A \sqsubseteq C$	$A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
concept definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
number restrictions	($\geq n$ R)	$\{d_1 \in \Delta \mid \#\{d_2 \in \Delta: (d_1, d_2) \in R^{\mathcal{I}}\} \geq n\}$
	($\geq n$ R)	$\{d_1 \in \Delta \mid \#\{d_2 \in \Delta: (d_1, d_2) \in R^{\mathcal{I}}\} \leq n\}$

Hereafter, for the sake of brevity, we will formalize examples by adopting DL syntax instead of OWL DL. In the prototypical system we realized, DIG (Description logics Implementation Group) [13] is exploited. It is a syntactic variant of OWL DL but it is less verbose, and this is a good feature in mobile ad hoc contexts.

In a DL framework, an ontology $\mathcal{T}$ is a set of axioms in the form: $A \sqsubseteq D$ or $A \equiv D$ where *A* is an atomic concept and *D* is a generic $\mathcal{ALN}$ concept. Such ontologies are also called Terminological Boxes (TBox).

Given an ontology $\mathcal{T}$ and two generic concepts *C* and *D*, DL reasoners provide at least two basic standard reasoning services: concept *subsumption* and concept *satisfiability*. In a nutshell they can be defined as reported hereafter.

- **Concept subsumption**: $\mathcal{T} \models C \sqsubseteq D$. Check if *C* is more specific than (implies) *D* with respect to the information modeled in $\mathcal{T}$.
- **Concept satisfiability**: $\mathcal{T} \models C \sqsubseteq \bot$. Check if the information in *C* is not consistent with respect to the information modeled in $\mathcal{T}$.

In a generic matchmaking process, subsumption and satisfiability may be powerful tools in case a Boolean answer is needed. Suppose you have an ontology $\mathcal{T}$ modeling information related to resources available in a given mobile environment and resources capabilities are described with respect to such ontology. In case you have a resource description *C* and a request *D*, whenever $\mathcal{T} \models C \sqsubseteq D$ holds, resource features entail the ones requested by the user. On the other hand, $\mathcal{T} \models C \sqcap D \sqsubseteq \bot$ means that resource capabilities are not compatible with the request.

However, in more advanced scenarios, yes/no answers do not provide satisfactory results. Often a result explanation is required. In [7] **Concept Abduction Problem (CAP)** and **Concept Contraction Problem (CCP)** were introduced and defined as non-standard inferences for DLs. CAP allows to provide an explanation when subsumption does not hold. Given an ontology $\mathcal{T}$ and two concepts *C* and *D*, if $\mathcal{T} \models C \sqsubseteq D$ is *false* then we compute a concept *H* (for hypothesis) such that $\mathcal{T} \models C \sqcap H \sqsubseteq D$ is true. That is, *H* is a possible explanation about why resource characteristics do not imply requested ones or, in other words, *H* represents missing capabilities in the resource *C*, able to completely satisfy a request *D* with respect to the information modeled in $\mathcal{T}$. Actually, given a CAP, there are more than one valid solution, hence some minimality criteria have to be defined. We refer the interested reader to [14] for further details.

If the conjunction $C \sqcap D$ is unsatisfiable in the TBox $\mathcal{T}$ representing the ontology, *i.e.*, *C*, *D* are not compatible with each other, the requester can retract some requirements *G* (for *Give up*) in *D*, to obtain a concept *K* (for *Keep*) such that $K \sqcap C$ is satisfiable in $\mathcal{T}$ (Concept Contraction Problem). CCP is formally defined as follows**.** Let $\mathcal{L}$ be a DL, *C*, *D* be two concepts in $\mathcal{L}$ and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both *C* and *D* are satisfiable in $\mathcal{T}$. A *Concept Contraction Problem* (CCP), identified by $\langle \mathcal{L}, C, D, \mathcal{T} \rangle$ is finding a pair of concepts $\langle G, K \rangle \in \mathcal{L} \times \mathcal{L}$ such that $\mathcal{T} \models D$

$\equiv G \sqcap K$, and $K \sqcap C$ is satisfiable in $\mathcal{T}$. Then K is a contraction of D according to C and $\mathcal{T}$.

If nothing can be kept in D during the contraction process, we get the worst solution – from a matchmaking standpoint – $\langle G, K\rangle = \langle D, \top\rangle$, that is give up everything of D. If $D \sqcap C$ is satisfiable in $\mathcal{T}$, that is a potential match occurs, nothing has to be given up and the solution is $\langle \top, D\rangle$, *i.e.*, give up nothing. Hence, a Concept Contraction problem amounts to an extension of satisfiability. Like for the abduction problem, some minimality criteria in the contraction must be defined [7], since usually one wants to give up as few things as possible.

In most cases, a pure logic-based approach could not be sufficient to decide between what to give up and what to keep. There is the need to define and use some extra-logical information. For instance, one could be interested in contracting only some specific part of a request, while considering the other ones as strictly needed [15].

A further interesting feature is the exploitation of previously described inference services with respect to an open world semantics scenario. Consider that, actually, if the provider specifies information about a resource which is not in the user request, this information is not used in the matchmaking process. That is, the so called *bonuses* put at disposal by a provider have no weight while retrieving appealing resources. On the other hand, if in the resource description there is no bonus, we conclude the information modeling the request implies the provided one [16]. If such bonuses are canceled from D, the implication relation is reached. Equivalently, the same relation holds if we add the bonuses to C. In the first case, removing bonuses from D, we basically produce a resource underspecification; in the latter one we have a query enrichment based on information which are elicited from D. Hence, a bonus can be seen as what has to be hypothesized in C, in order to make D implied by C, which may lead to an actual exact match. In DL words, when an inconsistency between a request and a resource description ensues, the only way to conclude the matchmaking process is by contracting C and subsequently continuing using only K_C, that is the part of C which is compatible with D. So, a slight extension of the approach outlined before allows us to consider bonuses to try reaching the equivalence relation between the request and the offered resource. Solving the CAP $\langle \mathcal{L}, P, K_C, \mathcal{T}\rangle$, where $\mathcal{T}$ is the reference domain ontology, we produce H intended as what has to be hypothesized and added to K_C to obtain $K_C \sqcap H \sqsubseteq D$. In this case H, from now on ***B*** (for **Bonus**), is the set of bonuses offered by D. By definition it results both $K_C \sqcap B \not\equiv \bot$ and $C \sqcap B \equiv \bot$.

Trivially, also, in this case, minimality in the hypotheses allows to avoid redundancy. In [7], among others, the conjunctive minimal solution to a CAP is proposed for DLs admitting a normal form with conjunctions of concepts. It is in the form $B = \sqcap_{i=1,\ldots,k} b_i$, where b_i are DLs concepts and the "$\sqcap$" is **irreducible**, *i.e.*, B is such that for each $h \in 1, \ldots, k$, $\sqcap_{i=1,\ldots,h-1,h+1,\ldots,k} b_i$ is not a solution for the CAP.

By applying the algorithm for bonuses computation, the returned set contains all the bonuses available in the resources (which can be used to refine the query) and what is still missing for each available resource to obtain an exact bidirectional match.

IV. Proposed Framework

A. Architecture

Fig. 1 shows the system architecture. A classical client/server paradigm is adopted: in our current prototype the resource provider is a fixed host over the Internet, exposing an enhanced DIG interface; the mobile client is connected through wireless technologies, such as IEEE 802.11 or UMTS/CDMA.

Available resources (supplies) were collected from several sources. The *DBpedia* [17] RDF Knowledge Base, which is an extract of structured information from Wikipedia, was used to automatically obtain relevant information for many entries. DBpedia is a prominent example of the Linked Data effort [18], aimed at publishing structured data on the Web and to connect data between different data sources. URIs (Uniform Resource Identifiers) and RDF (Resource Description Framework) provide the framework that allows both data to be machine understandable and related concepts from different datasets to be related to each other. Tens of datasets are already available, collectively containing several billion RDF statements and covering multiple application domains such as: encyclopedic, artistic and literary topics; healthcare, environmental and governmental data and statistics; commerce and finance. Resource providers can build innovative solutions, like the one presented here, upon these public Knowledge Bases (KBs).

RDF documents concerning resources of interest were directly retrieved from the KB using SPARQL query language. Obtained profiles were then sanitized (*e.g.*, by removing textual abstracts, redundant and unnecessary information) and aligned through a semi-automatic procedure to custom ontology (in the proposed case study it is referred to the cultural heritage domain). Then each semantic annotation was geographically tagged exploiting the Google Maps API. In the current prototype, each resource is supplied with a picture and a textual description. Finally, all resources were stored into a semantic registry whose records contain:

- A semantic annotation (in DIG language);
- A numeric ontology identifier, marking the domain ontology the annotation refers to;

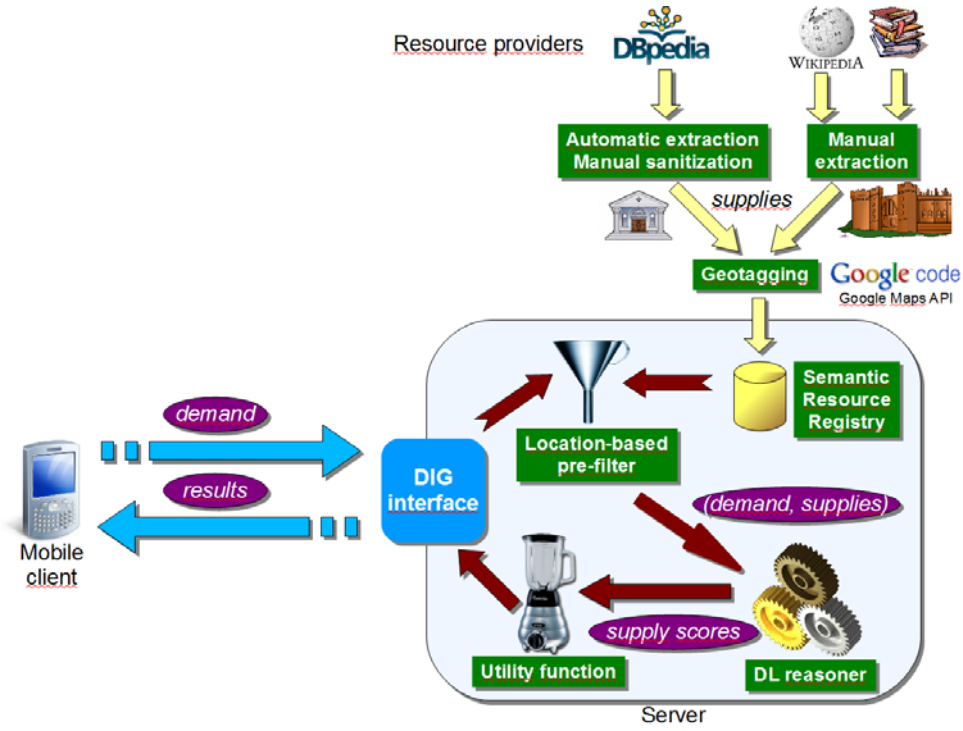


Figure 1 Architecture of the system prototype.

- A set of data-oriented attributes manageable by proper utility functions (see later on for further details);
- A set of user-oriented attributes.

On the client side, the user focuses on a given scenario early selecting the reference terminology. So she determines a specific context for the following interactions with the system. Different sessions in the application exploitation could refer to different ontologies and then could entail interactions aimed at different purposes. For example, a generic user could exploit the application as a pocket virtual guide for tourist purposes selecting a cultural heritage ontology and in a further phase, after concluding her visit, she can adopt it as a mobile shopping assistant to buy goods in a B2C (Business to Consumer) mobile marketplace: in that case she will select an e-commerce ontology.

Matchmaking can be carried out only among requests and supplied resources sharing the semantics of descriptions, *i.e.*, referred to the same ontology. Hence a preliminary agreement between client and server is required. Ontology identifiers are used for this purpose [19]. Then the client can submit her request, which consists in: (i) a DIG expression of the required resource features; (ii) geographical coordinates of the current device location; (iii) maximum acceptable distance for service/resource fruition.

When a request is received, the server performs the following processing steps.

1. Resources referring to the same ontology are extracted from the registry.
2. A location-based pre-filter excludes resources outside the maximum range w.r.t. the request, as explained in the following subsection.
3. The reasoning engine computes the semantic distance between request and each resource in range.
4. Results of semantic matchmaking are transferred to the utility function calculation module, which computes the final ranking according to the scoring functions reported hereafter.
5. Finally, the ranked list of best resource records is sent back to the client in a DIG reply.

B. *Location-based Resource Filtering*

Semantic-based matchmaking should be extended to take location into account, so as to provide an overall match degree that best suits the user needs in her current situation. Research in logic-based matchmaking has achieved some degree of success in extending useful inference services to DLs with concrete domains (datatype properties in Semantic Web words) [5], nevertheless these results are hardly transferred to mobile scenarios due to architectural and performance limitations. A different approach to the multi-attribute resource ranking problem is based on utility functions, a.k.a. Score Combination Functions (SCF). It consists in combining semantic-based match metrics with other partial scores computed from quantitative –in our case location-dependent– resource attributes.

In general, if a request and available resources are characterized by m attributes, the problem is to find a ranking of the set R of supplied resources according to the request $d=(d_1,d_2,\ldots,d_m)$. For each resource $r_i=(r_{i,1},r_{i,2},\ldots,r_{i,m}) \in R, 1 \le i \le |R|$, a set of local scores

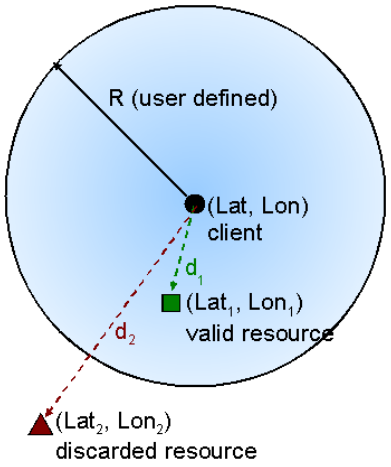


Figure 2 Location-based resource pre-filtering.

$s_{i,j}, 1 \le j \le m$ is computed as $s_{i,j} = f_j(d_j, r_{i,j})$. Then the overall score s_i for r_i is obtained by applying an SCF f, that is $s_i = f(s_{i,1}, s_{i,2}, \ldots, s_{i,m})$. Resources are so sorted and ranking is induced by the SCF.

The framework devised in this paper integrates a semantic score f_{ss} and a geographic score f_{gs}, combined by the SCF f_{sc}. The operating principle is illustrated in Fig. 2: a circular area is identified, centered in the user's position; the service provider will only return resources located in it. The user request contains a (latitude, longitude) pair of geographical coordinates for current device location along with a maximum range R. In the same way, each available resource collected by the provider is endowed with its coordinates. Distance d is computed between the user and the resource. If $d > R$, the resource is excluded, otherwise it is admitted to next processing stage.

The semantic score is computed as:

$$f_{ss}(r,s) = \frac{s\_match(r,s)}{\max(s\_match)}$$

where $s\_match(r,s)$ is the semantic match distance from request r to resource s (computed by means of the inference services explained before), while $\max(s\_match) \doteq s\_match(r, \top)$ is the maximum semantic distance, which depends on axioms in the reference domain ontology. Hence, $f_{SS} \in [0,1]$ and lower values are better.

The second score involves the physical distance:

$$f_{gs}(d) = \frac{d}{R}$$

Also, $f_{GS} \in [0,1]$ and lower values are preferable. It should be noticed that, in both local scoring functions, denominators are maximum values directly depending on the specific user request. They may change across different resource retrieval sessions, but correctly rank resources w.r.t. the request within the same session.

Finally, the SCF is defined as:

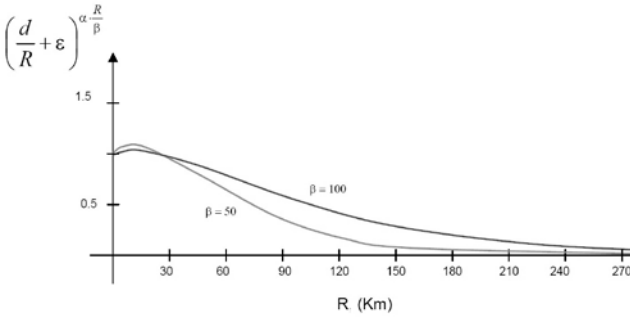


Figure 3 Geographic score contribution w.r.t. range R

$$f_{sc}(d,S) = 100 \cdot [1 - (f_{gs} + \varepsilon)^{\alpha\frac{R}{\beta}} \cdot (f_{ss} + \gamma)^{1-\alpha}]$$

It is a monotonic function providing a consistent resource ranking, and it converts results to a more user-friendly scale where higher outcomes represent better results. A tuning phase can be performed to determine parameter values following requirements of a specific discovery application. In detail, $\alpha \in [0,1]$ weighs the relevance of semantic and geographic factors, respectively. With $\alpha \to 0$ the semantic score is privileged, whereas with $\alpha \to 1$ the geographic one is made more significant. The exponent of the geographic factor is multiplied by $\frac{R}{\beta}$. This is because, when the maximum search range R grows, distance should reasonably become a more selective attribute, giving more relevance to resources in the user's neighborhood. The coefficient β regulates the curve decay, as shown in Fig. 3 for different values of β and $\alpha = 0.5$, $\varepsilon = 0$, $d = 30$ km.

Parameters $\varepsilon \in [0,1]$ and $\gamma \in [0,1]$ control the outcome in case of either semantic or geographic full match. As explained in Section III, semantic full match occurs when all features in the request are satisfied by the resource. Geographic full match occurs when the user is located exactly in the same place of resource she is looking for. Both cases are desirable but very unlikely in practical scenarios. Hence, in the model adopted for system evaluation we could pose $\varepsilon = \gamma = 0$:

$$f_{sc}(d,S) = 100 \cdot [1 - (f_{gs})^{\alpha\frac{R}{\beta}} \cdot (f_{ss})^{1-\alpha}]$$

This means that full matches will always be shown at the top of the result list, since either $f_{gs} = 0$ or $f_{ss} = 0$ implies $f_{sc} = 100$ regardless of the other factors.

V. System Prototype

A. Design and Development Guidelines

The above framework has been exploited within a prototypical mobile client for semantic-based service/resource discovery. It is aimed at employing the semantic matchmaking approach outlined above. Design and

development of the proposed application were driven by the following guidelines, taking the objective of maximizing efficiency, effectiveness and usability.

- Limited computing resources of the target platform must be carefully taken into account. From a performance standpoint, it is impractical to reuse existing Semantic Web tools and libraries on current mobile devices. A compact and optimized implementation of the required features and technologies is thus needed.
- Mobile computing platforms are much more heterogeneous than personal computers, with devices highly differentiating in form factor, computational and communication capabilities and operating systems. Cross-platform runtime environments can allow to overcome this fragmentation. This constraint can be partially in conflict with the former one, since a high-level platform increases portability (abstracting from hardware and operating system) usually at detriment of performance.
- Human-Computer Interaction (HCI) design should endorse the peculiarities of mobile and pervasive computing. Unlike their desktop counterparts, mobile applications are characterized by a bursting usage pattern, *i.e.*, with frequent and short sessions. Hence, a mobile Graphical User Interface (GUI) must be designed so that users can satisfy their needs in a quick and straightforward way. A task-oriented and consistent look and feel is required, leveraging familiar metaphors which most users are accustomed to.
- Finally, software design must be conscious of the inherent constraints of mobile ad-hoc networks. From the application perspective, the most important issues are unpredictable disconnections and low data rates. The former is mainly due to host mobility, higher transmission error rates of wireless links and limited battery duration. The latter is a typical concern of wireless networks with respect to wired ones and it is also due to energy saving requirements for small devices. Applications must be designed with built-in resilience against failures and QoS (Quality of Service) degradation at the network level, so as to prevent unexpected behaviors.

B. Technologies

For a greater compatibility with various mobile platforms, our client tool was developed using Java Micro Edition (ME) technology. Java ME is the most widespread cross-platform mobile environment and it offers a rich feature set. In general, the compliance with one of the Java ME profiles ensures the compatibility with a broad class of mobile computing devices. The Java Mobile Information Device Profile (MIDP) was selected, which is currently available for the majority of mobile phones and PDAs. Our tool is fully compliant with Java MIDP 2.0 specification and API. All UI elements are accessible through the keyboard/keypad of the mobile device; additionally, MIDP transparently adapts to pointer-based interaction (*e.g.*, via touchscreen) on platforms where it is available.

The MVC (Model-View-Controller) pattern was adopted in user interface design and two different GUI flavors were developed and evaluated. The UI has been carefully studied due to management and presentation of semantic-based data (ontology browsing and display of semantically annotated resource results), which have an intrinsically complex data model. The first GUI version was entirely based on MIDP API for the graphical interface, in order to maximize device compatibility and minimize application resource requirements. Custom items were built to extend the basic built-in GUI elements. The second version was entirely based on SVG (Scalable Vector Graphics) instead, using the Scalable 2D Vector Graphics API JSR-226. Vector-based graphics allows to produce better-looking graphics across screens with different resolutions. Furthermore, sophisticated animations and transition effects were introduced to make user interaction more pleasant and natural, as well as supporting intuitive user gestures for scrolling and dragging.

In order to allow location-based service/resource provisioning, the application exploits the Java Location API JSR-179 to determine the device's location. JSR-179 provides a unified API to interact with all location providers – *i.e.*, real-time positioning technologies – available on the device. These may include an internal GPS (Global Positioning System) receiver, an external GPS device connected *e.g.*, via Bluetooth and the mobile phone network itself (cell-based positioning). Accuracy depends on the positioning method, being typically higher for GPS than for cell-based techniques. Our tool requests a high-accuracy location determination firstly; if the accuracy requirement cannot be satisfied by available location providers on the device, the constraint is relaxed.

The proposed tool supports a subset of the DIG 1.1 interface extended for MaMaS-tng reasoner [20]. This HTTP-based interface allows interaction with the state-of-the-art of Knowledge Representation Systems (KRS) through a classical request/reply interaction.

A lightweight implementation of the client-side DIG interface has been developed in Java. A specialized library was designed for efficient manipulation of knowledge bases. In order to minimize runtime memory consumption, kXML Java streaming XML parser was adopted, which implements the open standard XML Pull API [21].

Streaming parsers allow an application to closely control the parsing process and do not build an in-memory syntax tree model for the XML document (as DOM parsers do). This increases speed and reduces memory requirements, which is highly desirable in resource-constrained environments. Moreover, streaming parsers are the best choice for processing XML data incoming from network connections, since parsing can be pipelined with the incoming input.

VI. CASE STUDY

Functional and non-functional features of the proposed system are motivated within a concrete case study in the cultural heritage tourism sector. Let us model the discovery problem as follows. *Jack is in Bari for business. He is keen on ancient architecture and after his last meeting, he is near the old town center with some spare time. He had never been in Bari before and he knows very little about the city. Being interested in medieval art and particularly in churches, he would like to visit interesting places near his current location. Under GPRS/UMTS or Wi-Fi coverage, his GPS-enabled smartphone can connect to a service/resource provider, in order to search for interesting items in the area.* The service provider keeps track of semantic annotations of touristic points of interest in Apulia region along with their position coordinates. The mobile application assists the user in the discovery process through the following three main tasks (depicted in Fig. 4).

Ontology management. *Firstly, Jack selects cultural heritage as the resource category he is interested in.* Different domain ontologies are used to describe general resource classes (*e.g.*, accommodation, cultural heritage, movie/theatre shows). At application startup, a selection screen is shown (Fig. 5), with a list of managed ontologies. Each Ontology is labeled by a Universally Unique IDentifier (OUUID), which allows an early agreement between user and provider. As explained in [19], this simple identification mechanism borrowed from the Bluetooth Service Discovery Protocol allows to perform a quick match between the ontologies managed by the user and by the provider also in case of mobile ad-hoc connections where users and providers are interconnected via wireless links (such as Bluetooth, 802.11, ZigBee and so on) and where a dependable Web link is unavailable. Anyway, in case the user cannot locally manage the chosen resource category, he can download the reference ontology either from near hosts or from the Web (when possible) exploiting the OUUID as reference identifier.

Semantic request composition. *Jack composes his semantic-based request through a fully visual form. He browses resource features modeled in the domain ontology*

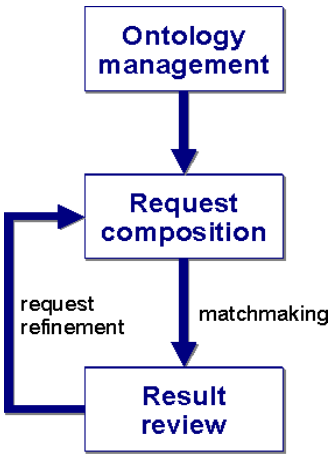


Figure 4 Tasks performed by the mobile client.

Figure 5 Ontology selection screen.

(partially reported in Table II for the sake of brevity) *and selects desired characteristics, without actually seeing anything of the underlying DL-based formalism. Then he submits his request.* Fig. 6 shows the ontology browsing screen. A scrollable list shows the current *focus* in the classification induced by terminological definitions and subsumptions. Directional keys of mobile device or swipe gestures on the touchscreen are used to browse the taxonomy by expanding an item or going back one level. Above the list, a *breadcrumb* control is displayed, so that the user can orient himself even in deeper ontologies. The tabs on top of the screen allow to switch from the Explore screen to the Request confirmation one (Fig. 7). There the user can remove previously selected features. Eventually, he specifies a retrieval diameter *R* and submits his request. Current prototype expresses the threshold in terms of distance, but a more intuitive indication clarifying if the user is on foot (possibly also specifying the terrain characteristics) or if he moves by car is also possible.

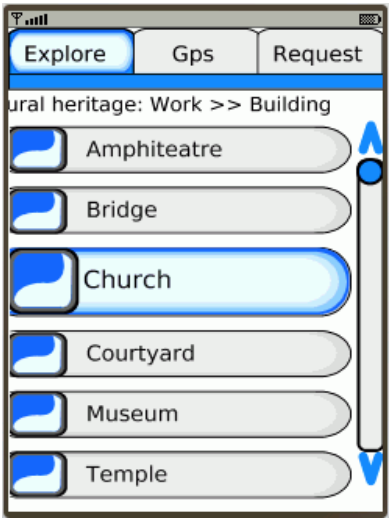


Figure 6 Ontology browsing screen.

TABLE II. EXCERPT OF AXIOMS IN THE CASE STUDY ONTOLOGY

AD ⊑ Age	BC ⊑ Age	Middle_Age ⊑ AD
Centralized ⊑ Floor_Plan	Longitudinal ⊑ Floor_Plan	Quadrangular ⊑ Floor_Plan
Square ⊑ Quadrangular	Byzantine ⊑ Style	Romanesque ⊑ Style
Gothic ⊑ Style	Baroque ⊑ Style	Portal ⊑ Architectural_Element
Cathedra ⊑ Architectural_Element	Aisle ⊑ Architectural_Element	Altar ⊑ Architectural_Element
Pulpit ⊑ Architectural_Element	Crypt ⊑ Architectural_Element	Apse ⊑ Architectural_Element
Window ⊑ ArchitecturalElement	Single_Light ⊑ Window	Double_Light ⊑ Window
Triple _Light ⊑ Window	Religious ⊑ Destination	Private ⊑ Destination
Public ⊑ Destination	Private ⊑ ¬Public	Private ⊑ ¬Religious
Building ⊑ ∃ has_age ⊓ ∃ has_floor_plan ⊓ ∃ has_style		
Residence ⊑ Building ⊓ ∃ Destination ⊓ ∀ Destination.Private		
Church ⊑ Building ⊓ ∃ Destination ⊓ ∀ Destination.Religious ⊓ ∃ has_altar ⊓ ∀ has_altar.Altar		
Castle ⊑ Residence		

Jack would like to visit a Romanesque Middle Age church with longitudinal floor plan and two aisles. W.r.t. the cultural heritage ontology, the request can be formally expressed as:

R: *Church ⊓ ∀has_age.Middle_Age ⊓ ∀ has_floor_plan.Longitudinal ⊓ ≥2 has_aisle ⊓ ∀ has_style.Romanesque*

It can be noticed that requests are formulated as DL conjunctive queries. Each conjunct is a requested resource feature; it can be an atomic concept selected from the ontology, a universal quantifier or an unqualified number restriction on roles. The GUI masks this level of complexity from the user, allowing him to simply browse lists of features and select the desired ones: translation into DL expression is automated, taking into account the concept

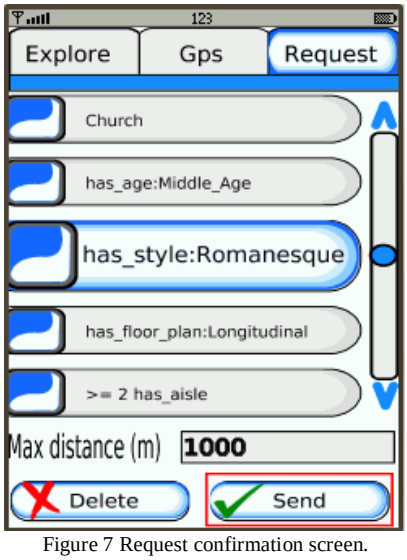


Figure 7 Request confirmation screen.

structure and relationships in the reference ontology.

The communication module was designed as a finite state machine to precisely retain knowledge about the progress of client-server interaction. By doing so, only failed operations are actually repeated, thus improving efficiency from both time and energy standpoints.

Results review and query refinement. *The server processes the request as explained in Section 4.* Let us consider the following resources in the provider KB:

S1*: Basilica of St. Nicholas, Bari (distance from user: d = 0.9 km). A Romanesque Middle Age church, with longitudinal floor plant, an apse, two aisles, three portals and two towers. Other notable elements are its crypt, altar, cathedra and Baroque ceiling. W.r.t. domain ontology, it is expressed as:*

Church ⊓=2 has_aisle ⊓ ∀has_age.Middle_Age ⊓ ∀ has_style.Romanesque ⊓=1 has_apse ⊓=3 has_portal ⊓=1 has_crypt ⊓=1 has_altar ⊓=2 has_tower ⊓=1 has_cathedra ⊓ ∃ ceiling_style ⊓ ∀ceiling_style.Baroque ⊓ ∀ has_floor_plan.Longitudinal

S2*: Norman-Hohenstaufen Castle, Bari (d = 0.57 km). It is described as a Middle Age castle, with Byzantine architectural style and a quadrangular plan with four towers. In DL notation:*

Castle ⊓ ∀has_floor_plan.Quadrangular ⊓=4 has_tower ⊓ ∀ has_style.Byzantine ⊓ ∀has_age.Middle_Age

S3*: Church of St. Scholastica (d = 1.3 km). It is described as a Romanesque Middle Age church, with longitudinal floor plan, three aisles, an apse and a tower. That is:*

Church ⊓ ∀has_style.Romanesque ⊓ ∀has_age.Middle_Age ⊓ ∀ has_floor_plan.Longitudinal ⊓=3 has_aisle ⊓=1 has_tower ⊓ =1 has_apse

S4*: Church of St. Mark of the Venetians, Bari (d = 0.65 km). It is described as a Romanesque Middle Age church with two single-light windows and a tower, whose DL translation is:*

Church ⊓ ∀has_style.Romanesque ⊓ ∀has_window.Single_Light ⊓ =2 has_window ⊓ ∀has_age.Middle_Age ⊓=1 has_tower

Table III reports on matchmaking results for the above example. **S3** is discarded in the location-based pre-filtering, as its distance from the user exceeds the limit, even though it would result in a full match. **S1** is a full match with the request, because it explicitly satisfies all user requirements. On the other hand, **S4** is described just as Romanesque Middle Age church, therefore due to OWA it is not specified whether it has a longitudinal floor plan with aisles or not: these characteristics become part of the *Hypothesis* computed through CAP. Finally, **S2** produces a partial match since it refers to a castle: this concept is incompatible with user request, so it forms the *Give Up* feature computed through CCP. Overall scores of advertised resources are finally computed. An example of result screen is reported in Fig. 8: retrieved resources are listed, best matching first. When the user selects a resource, its picture is shown as in Fig. 9 in addition to its address, distance from the user and semantically relevant properties contributing to the outcome.

If Jack is not satisfied with results, he can refine his request and submit it again. The user can go back to the ontology browsing screen to modify the request. Furthermore, he can select some elements of the *Bonus* (respectively *Give Up*) list in the result screen and they will be added to (resp. removed from) the request.

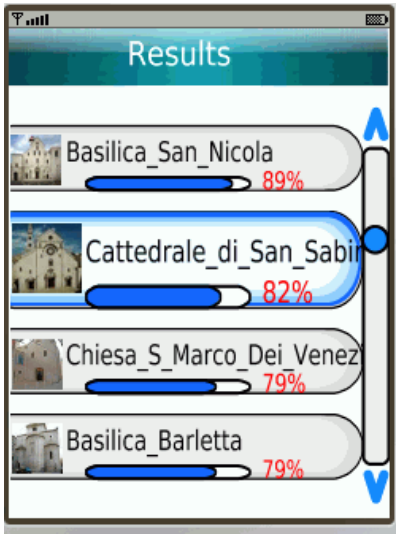


Figure 8 Displayed results.

VII. System Evaluation

A. System Performance

Performance analysis was executed on a Sony-Ericsson P990i smartphone, endowed with ARM processor at 208 MHz clock frequency, 64 MB of RAM, 80 MB of storage memory, a TFT 2,8" touchscreen with 240x320 pixel resolution, GSM/UMTS, IEEE 802.11b Wi-Fi connectivity and GPS, Symbian 9.1 operating system, manufacturer-supplied Java ME runtime compatible with MIDP 2.0 and all optional packages described in Section V-B. P990i smartphone was connected via UMTS to the matchmaking engine. Fig. 10 displays some screenshots of the u-tourism decision support system running on that mobile device. As performance metrics, RAM usage and latency time were considered for each tasks in Figure 4.

For memory analysis, the Memory Monitor profiling tool in Sun Java Wireless Toolkit was used. Results are reported in Fig. 11 for a typical usage session. RAM occupancy is always below 2 MB, which is the recommended threshold for MIDP applications. Memory peaks correspond to more graphical-intensive tasks, such as ontology browsing and preparation of the results screen.

TABLE III. Matchmaking Results

Supply	Match type	s_match [max=54]	Outcome	Score [α=0.5,β=1, γ=0.014, ε=0]
S1: Basilica of St. Nicholas	Full	0	Hypothesis H: ⊤ Bonus B: =1 has_apse ⊓ =3 has_portal ⊓ =1 has_crypt ⊓ =1 has_altar ⊓ =2 has_tower ⊓ =1 has_cathedra ⊓ ∃ ceiling_style ⊓ ∀ ceiling style.Baroque	88.8
S4: Church of St. Mark	Potential	3	Hypothesis H: ≥2 has_aisle ⊓ ∀ has_floor_plan. Longitudinal Bonus B: =1 has_tower ⊓ =2 has_window ⊓ ∀has_window.Single_Light	78.3
S2: Norman-Hohenstaufen Castle	Partial	11	Give up G: Church Keep K: Building ⊓ ∀has_age.Middle_age Hypothesis H: ∀has_floor_plan.Longitudinal ⊓ ≥2 has_aisle ⊓ ∀has_style.Romanesque Bonus B: =4 has_tower ⊓ ∀has_style.Byzantine	64.8
S3: Church of St. Scholastica	N.A.	N.A.	Discarded due to distance	N.A.

Figure 9 Result details screen.

The diagram in Fig. 11 is not significant for assessment of user-perceived latency, since idle times due to user reading the screen are also counted. Latency was measured through timers in the application code. The usage session described in the case study was repeated three times, exactly in the same way. Table IV contains average times obtained in loading each screen. The result list screen loading time includes interaction with the matchmaker (submitting the request, waiting for matchmaking computation, receiving the reply and building the result list GUI), and it is by far the highest value, posing a potential issue for practical usability. Latency in other tasks can be deemed as acceptable. In order to provide further insight into matchmaking computation performance – a key aspect for the feasibility of the proposed approach – a simulated testbed was used to assess semantic matchmaking processing times. Three ontologies with an increasing complexity were created and examined, and five different requests for each one were submitted to the system. Average response times were recorded. In Fig. 12 the matchmaking time (absolute and relative) is reported. The relative value is obtained by weighting the absolute matchmaking time according to the *ontology size* (expressed in terms of number of contained concepts). The relative time computation is needed because reasoning procedures are strongly conditioned by the complexity of the exploited ontology. So, the relative matchmaking evaluation produces an average *time per concept* which is a more precise indication of the matchmaking computational load with respect to the absolute one.

By examining the provided Fig. 12, it can be concluded that, for the most complex ontologies, semantic matchmaking time assumes a considerable value. Nevertheless, considering that applications as the one proposed here are required to be interactive and with fast response times, this is a relevant issue to solve. It was also pointed out by Ben Mokhtar *et al.* [21], who devised optimizations to reduce online reasoning time in a semantic-based mobile service discovery protocol. The main

Figure 10 Screenshots of prototype tool.

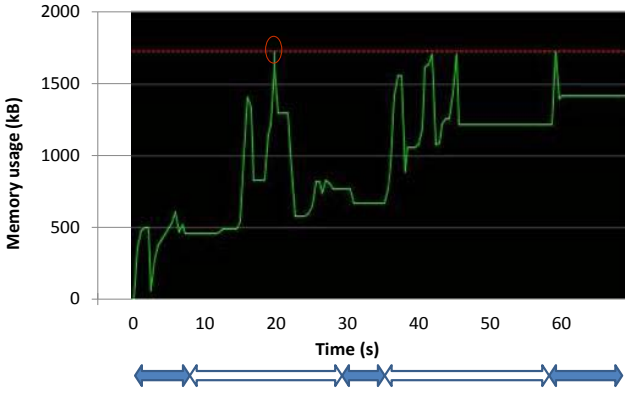
proposed optimizations were offline pre-classification of ontology concepts and concept encoding: both solutions, however, are viable in matchmaking schemes based on pure subsumption (and therefore able to provide only binary yes/no answers), but they are not directly applicable to our matchmaking approach.

B. *Semantic Web Technologies in Ubiquitous Computing*

Common issues rising from the integration of Semantic Web approaches with ubiquitous computing scenarios were evidenced in [23]. Let us take them as a check-list and evaluate our proposal against it.

TABLE IV. GUI LATENCY.

Loading Screen	Loading latency (s)
Ontology selection	0.678
Ontology browsing	3.136
Request confirmation	0.939
Result list	11.107



Task	Ontology selection	Ontology browsing	Request confirmation	Result calculation and review	Result details
Memory Peak (kB)	888	1846	690	1651	1459

Figure 11 Main memory usage profile and peaks.

A. *Simple architectures lack intelligence of Semantic Web technologies.* The current proposal allows mobile devices equipped with commonly available technologies to fully exploit semantic-based resource discovery. Ideas and technologies devised for resource retrieval in the Semantic Web were adapted with a satisfactory success, through careful selection of features and optimization of implementation.

B. *Semantic Web architectures use devices with a secondary, passive role.* In our prototype the client has a key role and it does not only act as a GUI for request composition via ontology browsing. It also enables: location determination; interaction with a state-of-the-art, DIG-based reasoning engine; interactive visualization of discovery results for query refinement.

C. *Semantic Web architectures rely on a central component that must be deployed and configured beforehand for each specific scenario.* The proposed system prototype still relies on a centralized server for resource matchmaking. Future work aims at building a fully mobile peer-to-peer architecture. A major step is to design and implement embedded DL reasoners with acceptable performance: early results have been achieved in this concern [24].

D. *Most architectures do not use the Web communication model, essentially HTTP.* For communication we only use DIG, a standard based on the HTTP POST method and on an XML-based concept language. Such a choice allows – among other things – to cope with scalability issues: particularly, the interaction model is borrowed from the Web experience in order to grant an acceptable behaviour also in presence of large amounts of exchanged data.

E. *Devices are not first-class actors in the environment with autonomy, context-awareness and reactiveness.* Though the typical usage scenario for our current prototype is user-driven, it shows how a non-technical user can fully leverage Semantic Web technologies via her personal mobile device to discover interesting resources in her surroundings.

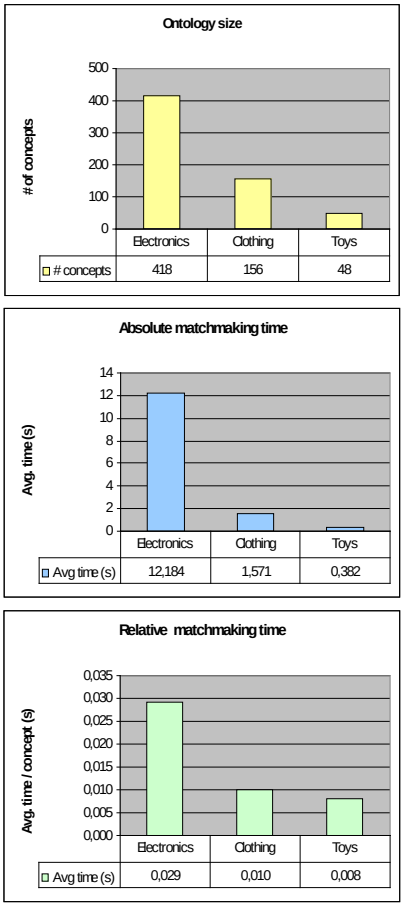


Figure 12 Performace evaluation of semantic matchmaking.

VIII. Related Work

Significant research and industry efforts are focusing on service/resource discovery in mobile and ubiquitous computing. The main challenge is to provide paradigms and techniques that are effective and flexible, yet intuitive enough to be of practical interest for a potentially wide user base.

In [25], a prototypical mobile client is presented for semantic-based mobile service discovery. An adaptive graph-based representation allows OWL ontology browsing. However, a large screen seems to be required to explore ontologies of moderate complexity with reasonable comfort. Also preference specification requires a rather long interaction process, which could be impractical in mobile scenarios. Authors acknowledged these issues and introduced heuristic mechanisms to simplify interaction, *e.g.*, the adoption of default values.

In [26], a location- and context-aware mobile Semantic Web client is proposed for tourism scenarios. The goal of integrating multiple information domains has led to a division of the user interface into many small sections,

whose clarity and practical usability seem questionable. Moreover, knowledge is extracted from several independent sources to build a centralized RDF triple store accessible through the Internet. The proposed architecture is therefore hardly adaptable to mobile ad-hoc environments.

A more open framework is presented in [27], allowing the translation and publication of OpenStreetMap data into an Open Linked Data repository in RDF. A public endpoint on the Web allows users to submit queries in SPARQL RDF query language, in order to retrieve geo-data of a specific region, optionally filtered by property values. Nevertheless, developed facilities currently cannot support advanced LBSs such as semantic matchmaking for resource discovery.

Van Aart *et al.* [28] presented a mobile application for location-aware semantic search, bearing some similarities with the proposal presented here. An augmented reality client for iPhone sends GPS position and heading to a server and receives an RDF dataset relevant to locations and objects in the route of the user. Applicability of the approach is limited by the availability of pre-existing RDF datasets, since the problem of creating and maintaining them was not considered.

DBpedia Mobile [29] allows user to search for resources located nearby, by means of information extracted from DBpedia and other datasets. The system also enables users to publish pictures and reviews that further enrich POIs. The user may filter the map for resources matching specific constraints or a SPARQL query. However, in the first case approximated matches are not allowed; a resource is found if and only if the overall query is satisfied. In the second case, the SPARQL query builder requires the user to know language fundamentals. Our approach aims at overcoming both restrictions.

Peer-to-peer interaction paradigms are actually needed for fully decentralized semantic-based discovery infrastructures. Hence, mobile hosts themselves should be endowed with reasoning capabilities. Pocket KRHyper [30] was the first available reasoning engine for mobile devices. It provides satisfiability and subsumption inference services, which have been exploited by authors in a DL-based matchmaking between user profiles and descriptions of resources/services [31]. A limitation of that prototype is that it does not allow explicit explanation of outcomes. More recently, in [24] an embedded DL reasoning engine was presented in a mobile dating application, though applicable to other discovery scenarios. It acts as a mobile semantic matchmaker, exploiting non-standard inference services also used in the present framework. Semantically annotated personal profiles are exchanged via Bluetooth and matched with preferences of mobile phone users, to discover suitable partners in the neighbourhood.

Due to the resource constraints of mobile devices, as well as to the choice of a cross-platform runtime environment, both the above solutions privilege simplicity of managed resource/service descriptions over expressiveness and flexibility. We conjecture that a native language optimized implementation can provide acceptable performance for larger ontologies and more resource-intensive inferences.

IX. Conclusion and Future Work

The paper presented a framework for semantic-enabled resource discovery in ubiquitous computing. It has been implemented in a visual mobile DSS able to retrieve resources/services through a fully dynamic wireless infrastructure, without relying on support facilities provided by wired information systems. The system recognizes via GPS the user location and grades matchmaking outcomes according to proximity criteria. Future work aims at simplifying the complexity of matchmaker module claiming for optimization and rationalization of the reasoner structure, in order to improve performance and scalability and to allow its integration into mobile computing devices and systems. Furthermore, a navigation engine will be integrated in the mobile application and the user interface will be enhanced to be even more friendly for non-expert users. Finally, we are investigating a new approach based on the semantic-based annotation of OpenStreetMap cartographic data, in order to exploit crowd-sourcing to face the issue of resource annotation.

Acknowledgment

The authors acknowledge partial support of Apulia region strategic projects PS_025 and PS_121.

References

[1] M. Ruta, F. Scioscia, E. Di Sciascio, G. Piscitelli, “Semantic-based Geographical Matchmaking in Ubiquitous Computing”, *The Fourth International Conference on Advances in Semantic Processing (SEMAPRO 2010)*, pp. 166-172, 2010.

[2] A. Smith, “35% of American adults own a smartphone”, Pew Internet & American Life Project, July 2011, http://pewinternet.org/~/media//Files/Reports/2011/PIP_Smartphones.pdf, accessed on July 12, 2011.

[3] T. Di Noia, E. Di Sciascio, F.M. Donini, M. Ruta, F. Scioscia, E. Tinelli, “Semantic-based Bluetooth-RFID interaction for advanced resource discovery in pervasive contexts”. *International Journal on Semantic Web and Information Systems*, vol. 4(1), pp. 50-74, 2008.

[4] A. Langegger, W. Wöß, “Product finding on the semantic web: A search agent supporting products with limited availability”. *International Journal of Web Information Systems*, Vol. 3, No. 1/2, Emerald Group Publishing Limited, 2007, pp. 61-88.

[5] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, P. Patel-Schneider, “The Description Logic Handbook”, Cambridge University Press, New York, 2002.

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

[7] T. Di Noia, E. Di Sciascio, F.M. Donini, “Semantic matchmaking as non-monotonic reasoning: A description logic approach”. *Journal of Artificial Intelligence Research*, vol. 29(1), pp. 269-307, AAI, 2007.

[8] D.L. McGuinness, F. van Harmelen, "OWL Web Ontology Language", W3C Recommendation. http://www.w3.org/TR/owl-features/, accessed on July 12, 2011.

[9] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", Scientific American, vol. 248(4), pp.34–43, 201.

[10] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara. "Semantic Matching of Web Services Capabilities". *The Semantic Web - ISWC 2002*, Lecture Notes in Computer Science, vol. 2342, pp. 333-347, 2002.

[11] L. Li, I. Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology". *International Journal of Electronic Commerce*, vol. 8(4), pp. 39–60, 2004.

[12] R. Brachman, H. Levesque, "The Tractability of Subsumption in Frame-based Description Languages". *In proc. of Fourth National Conference on Artificial Intelligence (AAAI-84)*, pp. 34-37, Morgan Kaufmann, 1984.

[13] S. Bechhofer, R. Möller, P. Crowther, "The DIG Description Logic Interface". *In proc. of the 16th International Workshop on Description Logics (DL'03)*, CEUR Workshop Proceedings, vol. 81, 2003.

[14] S. Colucci, T. Di Noia, A. Pinto, A. Ragone, M. Ruta, E. Tinelli, "A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces". *International Journal of Electronic Commerce*, vol. 12(2), pp. 127-154, 2007.

[15] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, 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(4), pp. 345-361, 2005.

[16] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, A. Ragone, "Knowledge Elicitation for Query Refinement in a Semantic-Enabled E-Marketplace". *In proc. of 7th International Conference on Electronic Commerce (ICEC05)*, pp. 685-691, ACM Press, 2005.

[17] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, "Dbpedia: A nucleus for a web of open data", *The Semantic Web*, pp. 722-735, Springer, 2007.

[18] C. Bizer, T. Heath, K. Idehen, T. Berners-Lee, "Linked data on the web". *In proc. of the 17th international conference on World Wide Web*, pp. 1265-1266, ACM, 2008.

[19] M. Ruta, T. Di Noia, E. Di Sciascio, F.M. Donini, "Semantic based collaborative p2p in ubiquitous computing". *Web Intelligence and Agent Systems*, vol. 5, n. 4, pp. 375–391, 2007.

[20] T. Di Noia, E. Di Sciascio, F.M. Donini, M. Mongiello, "A System for Principled Matchmaking in an Electronic Marketplace". *International Journal of Electronic Commerce*, vol. 8(4), pp. 9-37, 2004.

[21] A. Slominski, S. Haustein, "XML Pull Parsing API", 2005, available at http://xmlpull.org/, accessed on July 12, 2011.

[22] S. Ben Mokhtar, A. Kaul, N. Georgantas, V. Issarny, "Efficient Semantic Service Discovery in Pervasive Computing Environments". *In proc. of the ACM/IFIP/USENIX 7th International Middleware Conference, Middleware '06*, 2006.

[23] J.I. Vazquez, D. López de Ipiña, I. Sedano, "SoaM: A Web-powered Architecture for Designing and Deploying Pervasive Semantic Devices". *International Journal of Web Information Systems*, vol. 2(3/4) , pp. 212-224, Emerald Group Publishing Limited, 2006.

[24] M. Ruta, T. Di Noia, E. Di Sciascio, F. Scioscia, "Abduction and Contraction for Semantic-based Mobile Dating in P2P Environments". *In proc. of 6th IEEE/WIC/ACM International Conference on Web Intelligence (WI08)*, pp. 626–632, IEEE, 2008.

[25] O. Noppens, M. Luther, T. Liebig, M. Wagner, M. Paolucci, "Ontology supported Preference Handling for Mobile Music Selection". *In proc. of the Multidisciplinary Workshop on Advances in Preference Handling*, Riva del Garda, Italy, 2006.

[26] M. Wilson, A. Russell, D. Smith, A. Owens, M. Schraefel, "mSpace Mobile: A Mobile Application for the Semantic Web." *In proc. of the End User Semantic Web Workshop at ISWC 2005*, 2005.

[27] S. Auer, J. Lehmann, S. Hellmann. LinkedGeoData - Adding a Spatial Dimension to the Web of Data. *In proc. of 8th International Semantic Web Conference (ISWC). Springer-Verlag*, 2009.

[28] C. J. van Aart, B. J. Wielinga, and W. R. van Hage. Mobile Cultural Heritage Guide: Location-Aware Semantic Search. *In proc. of 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010)*, volume 6385 of Lecture Notes in Computer Science, pages 257–271, Berlin Heidelberg, 2010. Springer-Verlag.

[29] C. Becker, C. Bizer, "Exploring the Geospatial Semantic Web with DBpedia Mobile". *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7(4), pp. 278-286, Elsevier, 2009.

[30] A. Sinner, T. Kleemann, "KRHyper - In Your Pocket". *In proc. of 20th International Conference on Automated Deduction (CADE-20)*, pp. 452–457, 2005.

[31] T. Kleemann, A. Sinner, "User Profiles and Matchmaking on Mobile Phones". *In proc. of 16th International Conference on Applications of Declarative Programming and Knowledge Management (INAP2005)*, pp. 135-147, Springer, 2005.