Semantic-based Geographical Matchmaking in Ubiquitous Computing

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*—A full exploitation of semantics in mobile environments enhances discovery effectiveness allowing the instant fruition of services and resources. Hence, nowadays ever increasing efforts are spent in making available tools able to exploit Semantic Web techniques and technologies also in ubiquitous computing. This paper presents a platform-independent mobile semantic discovery framework as well as a working prototypical implementation, which enables advanced knowledge-based services taking into account user's location. The proposed approach is explained and motivated in a ubiquitous tourism case study, where some early evaluations are presented to prove its feasibility and usefulness.**

Keywords-Semantic Web, Ubiquitous Computing, Location-based Services, Resource Discovery.

I. INTRODUCTION

Techniques and ideas of the Semantic Web initiative are potential means to give flexibility to discovery [1]. 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 and inference of new knowledge; (iii) validity of the Open World Assumption (OWA) (what is not specified has not to be interpreted as a constraint of absence) [2], 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 frameworks.

This paper presents a general approach for a semantic-based discovery in ubiquitous environments. It has been implemented in a Decision Support System (DSS), presented here with reference to a u-tourism (ubiquitous tourism) [3] case study. Users equipped with handheld devices can exploit semantic resource annotation to obtain a logic-based ranking and explanation of results, while all technicalities are hidden from them. Furthermore, a selective discovery is performed based on proximity measures. In the proposed touristic virtual guide application, this feature has been implemented by integrating a positioning module in the discovery tool. The application recognizes the user location and grades matchmaking outcomes according to geographical criteria, presenting an intuitive Graphical User Interface (GUI).

Main features of the proposed approach are: (i) semantic-based ranking of retrieved resources; (ii) full exploitation of non-standard inferences presented in [4] to enable refinement of user requests; (iii) fully graphical interface usable with no prior knowledge of logic principles; (iv) no physical space-temporal constraints in system exploitation. The interest domain is modeled with an OWL (Web Ontology Language) [5] ontology.

The remaining of the paper is structured as follows: in the next section, relevant background is revised; Section 3 describes framework and approach with the aid of the case study in Section 4. Some evaluations about the system are reported in Section 5 before concluding the paper.

II. BACKGROUND

The reader is supposed to be familiar with Description Logics (DLs), a family of logic formalisms for Knowledge Representation [2]. In this paper we will refer to the $\mathcal{ALN}$ (Attributive Language with Unqualified Number Restrictions) Description Logic, a subset of OWL DL having polynomial computational complexity for standard and non-standard inferences. Hereafter, for the sake of brevity, examples will be formalized by adopting DL syntax, whereas in our prototypes we exploit DIG (a syntactic variant of OWL) [6] because it is less verbose than OWL.

DL reasoners provide at least two basic standard inference services: concept *subsumption* (a.k.a. classification) and concept *satisfiability* (a.k.a. consistency) [2]. Given *R* (for Request) and *O* (for Offered resource), both consistent w.r.t. a common ontology $\mathcal{T}$ (containing axioms modeling domain knowledge), logic-based approaches to matchmaking in literature [7] use classification and consistency to grade match results in five categories: (i) *exact* - every feature requested in *R* is exactly the same provided by *O* and vice versa; (ii) *full-subsumption* - every feature requested in *R* is contained in *O*; (iii) *plug-in* - every feature offered in *O* is contained in *R;* (iv) *potential-intersection* - there is an intersection and no conflicts between the features offered in *O* and the ones requested in *R*; (v) *partial-disjoint* - some features requested in R are conflicting with some offered in *O*.

 ISBN: 978-1-61208-000-0

Exact and full matches are the best ones for requesters, but they are infrequent in practical scenarios. A sequence of potential and partial matches is more likely. In [8], *Concept Abduction Problem (CAP)* and *Concept Contraction Problem (CCP)* were introduced and defined as non standard inferences for DLs. They allow to compute a logic-based ranking of potential and partial matches best approximating the request. Furthermore, they provide explanation of matchmaking outcomes, which is highly desirable to justify results so increasing user confidence in the DSS.

A noticeable feature is the exploitation of the above inference services w.r.t. an Open World semantics. If *R* and *O* are in potential match, the characteristics *B* (for *bonus*) [9] specified in *O* but not requested in *R* represent knowledge that can be elicited and proposed to the requester in order to refine her initial query. *B* can be computed by solving a CAP [9]. The bonus characteristics represent information the user might not be aware of or she initially considers not relevant. Hence, they are very useful in a query refinement process.

A. Related Work

Significant research and industry efforts have been 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 [10], 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 [11] 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.

Peer-to-peer interaction paradigms are actually necessary for fully decentralized semantic-based discovery infrastructures. Hence, mobile hosts themselves should be endowed with reasoning capabilities. Pocket KRHyper [12] 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 [13]. A limitation of that prototype is that it does not allow explicit explanation of outcomes. More recently, in [14] 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 neighborhood.

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.

III. System Outline

Figure 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.

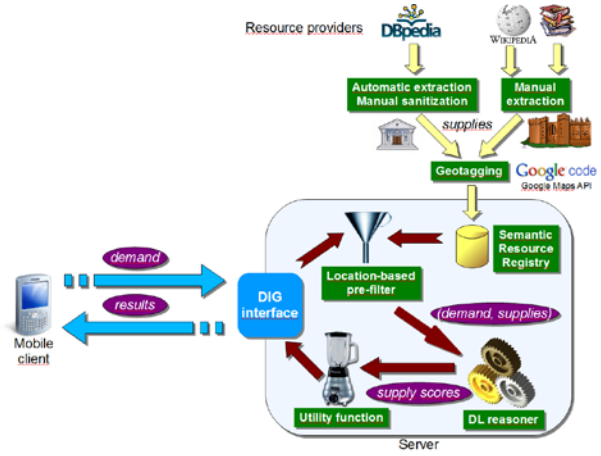

Figure 1. Architecture of the system prototype.

Available resources (*supplies*) were collected from several sources. The DBpedia RDF Knowledge Base (available at http://wiki.dbpedia.org), 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 [15], aimed at publishing structured data on the Web and to connect data between different data sources. RDF provides the semantic 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 diverse 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

 ISBN: 978-1-61208-000-0

language. Obtained profiles were then sanitized (e.g., by removing textual abstracts, redundant and unnecessary information) and aligned to custom ontology for the cultural heritage domain through a semi-automatic procedure. Then each semantic annotation was geographically tagged exploiting the Google Maps API. Finally, all resources were stored into a *semantic registry* whose records contain: (i) a semantic annotation (in DIG language); (ii) a numeric ontology identifier, marking the domain ontology the annotation refers to; (iii) a set of data-oriented attributes manageable by proper *utility functions* (see later on for further details) and depending on the specific application; particularly, the tool proposed here requires a *(latitude, longitude)* pair of geographical coordinates; (iv) a set of user-oriented attributes. In the current prototype, each resource is supplied with a picture and a textual description.

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.*, referring to the same ontology. Hence a preliminary agreement between client and server is required. Ontology identifiers are used for this purpose [16]. 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 subsection below. 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 in the next subsection. 5. Finally, the ranked list of best resource records is sent back to the client in a DIG reply.

A. *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) [2], nevertheless these results are hardly transferred to mobile scenarios due to 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* $s_{i,j}, 1 \le j \le m$ 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 s*emantic score* f_{ss} and a *geographic score* f_{gs}, combined by the SCF f_{sc}. The operating principle is illustrated in Figure 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.

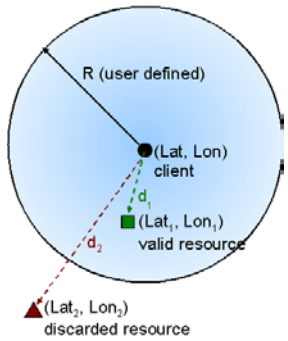

Figure 2. Location-based resource pre-filtering.

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:

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

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 $\alpha \to 0$ we privilege the semantic score, 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 immediate proximity. The coefficient β regulates the curve decay, as shown in Figure 3 for different values of β and $\alpha = 0.5$, $\varepsilon = 0$, $d = 30$ km.

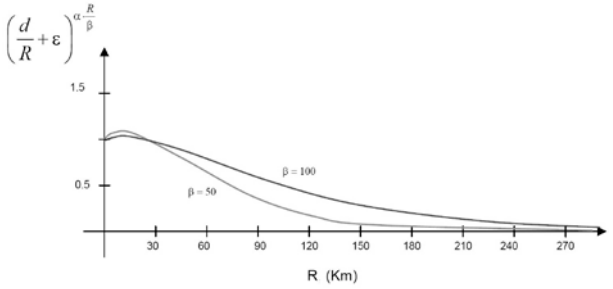

Figure 3. Geographic score contribution w.r.t. range R

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 II, 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.

B. *Design and development methodology*

Mobile computing devices are very heterogeneous in terms of screen size, input methods, computational and communication capabilities and operating systems. Furthermore, Human-Computer Interaction (HCI) design should endorse the peculiarities of handheld devices. Unlike their desktop counterparts, mobile applications are characterized by a *bursting* usage pattern, *i.e.*, with frequent and short sessions. Hence, a mobile 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 take into account the inherent constraints of mobile ad-hoc networks: from the application perspective, the most important issues are unpredictable disconnections and low data rates.

For a greater compatibility with various mobile platforms, our client tool was developed using Java Micro Edition (ME) technology. The Java Mobile Information Device Profile 2.0 (JSR 37, JSR 118, available at http://www.oracle.com/technetwork/java/index-jsp-138820.html) was selected. All UI elements are accessible either through the keyboard/keypad or the touchscreen of the mobile device.

The MVC (Model-View-Controller) pattern was adopted in user interface design. This was important for the management and presentation of semantic-based data, which have an intrinsically complex data model. The GUI was entirely based on SVG (Scalable Vector Graphics), using the Scalable 2D Vector Graphics for Java ME (JSR 226, available at http://jcp.org/en/jsr/detail?id=226).

The application exploits the Java Location API (JSR 179, available at http://jcp.org/en/jsr/detail?id=179) to allow location-based service/resource provisioning. It provides a unified API to interact with all *location providers –i.e.*, real-time positioning technologies– available on the device. These may include a GPS (Global Positioning System) receiver and the mobile phone network itself (cell-based positioning).

The proposed tool supports a subset of the DIG 1.1 interface extended for MaMaS-tng reasoner (see the MatchMaking Service, available at http://sisinfab.poliba.it/MAMAS-tng/). A lightweight implementation of the client-side DIG interface has been developed in Java. A specialized library was designed to manipulate Knowledge Bases (KBs). In order to minimize runtime memory usage, *kXML* (kXML 2, available at http://kxml.sourceforge.net/) Java streaming XML parser was adopted, which implements the open standard XML Pull API. Streaming parsers do not build an in-memory syntax tree model. They are also suitable for XML data incoming from network connections, since parsing can be pipelined with the incoming input.

 ISBN: 978-1-61208-000-0

IV. 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 centre 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 Figure 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 (Figure 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 [14], 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 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.

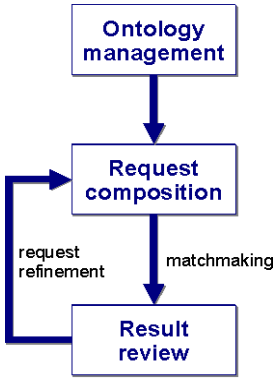

Figure 4. Tasks of the mobile client.

Figure 5. Ontology selection screen

Semantic request composition. *Jack composes his semantic-based request through a fully visual form. He browses resource features modeled in the domain ontology* (partially reported in Table I for the sake of brevity) *and selects desired characteristics, without actually seeing anything of the underlying DL-based formalism. Then he submits his request.*

Figure 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.

Figure 6. Ontology browsing screen

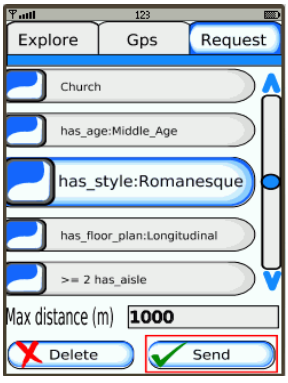

Figure 7. Request confirmation screen

TABLE I. 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		

The tabs on top of the screen allow user to switch from the Explore screen to the Request confirmation screen (Figure 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.

 ISBN: 978-1-61208-000-0

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 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 3.* 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, this 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.

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.

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.

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

Table II reports 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 with user request, since it refers to a castle: this concept is incompatible with user request, so it forms the *Give Up* feature computed through CCP.

TABLE II. 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.

Overall scores of advertised resources are finally computed. The result screen is reported in Figure 8: retrieved resources are listed, best matching first. When the user selects a resource, its picture is shown as in Figure 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

Figure 9. Result details screen

 ISBN: 978-1-61208-000-0

V. System Evaluation

Common issues rising from the integration of Semantic Web approaches with ubiquitous computing scenarios were evidenced in [17]. Let us take them as a check-list and evaluate our proposal against it.

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 implementations.

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 [14].

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 and 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 behavior 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.

VI. Conclusion and Future Work

A framework has been presented 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. It 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, the application user interface has to be enhanced and redesigned to be even more friendly for non-expert users. We are investigating a new approach directly and automatically browsing the DBpedia KB.

References

[1] Langegger, A. and Wöß W. “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.

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

[3] Watson, R., Akselsen, S., Monod, E., and Pitt, L. “The Open Tourism Consortium: Laying The Foundations for the Future of Tourism.” *European Management Journal*, Vol. 22, No. 3, 2004, pp. 315–326.

[4] Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., and Tinelli, E. “A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces”. *International Journal of Electronic Commerce*, Vol. 12, No. 2, 2007, pp. 127-154.

[5] McGuinness D.L. and van Harmelen F., editor. “OWL Web Ontology Language, W3C Recommendation, 2004. http://www.w3.org/TR/owlfeatures/”. Accessed 14 October 2008.

[6] Bechhofer, S., Möller, R., and Crowther, P. “The DIG Description Logic Interface.” In: *Proceedings of the 16th International Workshop on Description Logics (DL’03)*. Vol. 81 of *CEUR Workshop Proceedings*, 2003.

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

[8] Di Noia, T., Di Sciascio, E., Donini, F.M., and Mongiello, M. “Abductive matchmaking using description logics.” In: *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence IJCAI-03*, Acapulco, Mexico, MK, Vol. 18, 2003, pp. 337-342.

[9] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., and Ragone, A. “Knowledge Elicitation for Query Refinement in a Semantic-Enabled E-Marketplace”. In *7th International Conference on Electronic Commerce ICEC 05*, ACM Press, 2005, pp. 685-691.

[10] Noppens, O., Luther, M., Liebig, T., Wagner, M., and Paolucci, M.. “Ontology supported Preference Handling for Mobile Music Selection.” In: *Proceedings of the Multidisciplinary Workshop on Advances in Preference Handling*, Riva del Garda, Italy, 2006.

[11] Wilson, M., Russell, A., Smith, D., Owens, A., and Schraefel, M. “mSpace Mobile: A Mobile Application for the Semantic Web.” In: *Proceedings of the End User Semantic Web Workshop at ISWC 2005*, 2005.

[12] Sinner, A. and Kleemann, T. “KRHyper - In Your Pocket.” In: *Proc. of 20th International Conference on Automated Deduction (CADE-20)*, 2005, pp. 452–457.

[13] Kleemann, T. and Sinner, A. “User Profiles and Matchmaking on Mobile Phones.” In Bartenstein, O., ed.: *Proc. of 16th International Conference on Applications of Declarative Programming and Knowledge Management INAP2005*, 2005.

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

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

[16] Ruta, M., Di Noia, T., Di Sciascio, E., and Donini, F.M.. “Semantic based collaborative p2p in ubiquitous computing.” *Web Intelligence and Agent Systems*, Vol. 5, No. 4, 2007, pp. 375–391.

[17] Vazquez, J.I., López de Ipiña, D., and Sedano, I. “SoaM: A Web-powered Architecture for Designing and Deploying Pervasive Semantic Devices”. *International Journal of Web Information Systems*, Vol. 2, No. 3/4, Emerald Group Publishing Limited, 2006, pp. 212-224.

 ISBN: 978-1-61208-000-0