SHERLOCK: A System for Location-Based Services in Wireless Environments Using Semantics

Roberto Yus
University of Zaragoza
María de Luna 1
Zaragoza, Spain
ryus@unizar.es

Eduardo Mena
University of Zaragoza
María de Luna 1
Zaragoza, Spain
emena@unizar.es

Sergio Ilarri
University of Zaragoza
María de Luna 1
Zaragoza, Spain
silarri@unizar.es

Arantza Illarramendi
Basque Country University
Manuel de Lardizábal s/n
San Sebastián, Spain
a.illarramendi@ehu.es

ABSTRACT

Nowadays people are exposed to huge amounts of information that are generated continuously. However, current mobile applications, Web pages, and Location-Based Services (LBSs) are designed for specific scenarios and goals. In this demo we show the system SHERLOCK, which searches and shares up-to-date knowledge from nearby devices to relieve the user from knowing and managing such knowledge directly. Besides, the system guides the user in the process of selecting the service that best fits his/her needs in the given context.

Categories and Subject Descriptors

H.4.0 [**Information Systems Applications**]: General; H.2.4 [**Database Management**]: Systems—*Query Processing*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Semantic networks*

General Terms

Algorithms

Keywords

Location-based services, knowledge representation and reasoning, real-time query processing, mobile computing

1. INTRODUCTION

Nowadays people are exposed to huge amounts of information that are generated continuously; the ever-increasing use of mobile devices and their pervasiveness enable users to receive and create new contents almost anywhere. Currently, *Location-Based Services (LBSs)* [6] are very popular, as they offer users customized information taking into account their current location. However, whereas users have access to a wide variety of information, current mobile apps, Web pages, and LBSs are designed for specific scenarios and goals. So, the information about the context they manage is not explicitly represented but embedded in their code. Moreover, developing services ad hoc for specific purposes leads to the fact that there exist thousands of them (even with the same purpose), and therefore it is difficult to choose the most suitable one.

Imagine a WWW 2013 attendee that has just arrived in the Rio de Janeiro-Galeão International Airport, and needs a way to reach the conference hotel; this information could be obtained by visiting a tourist office, searching a local transportation website, or even downloading a mobile app[1]. After checking in, he could be interested in finding other nearby conference attendees to talk to them or even to go sightseeing (again, he should browse the Web to find information about interesting places to visit). Thus, it is the user himself who is in charge of knowing/finding the interesting and updated information sources and gathering and correlating all this information; even worse, he will have to know/find all these *updated* information about each city he would visit.

Some ad hoc solutions have been proposed to provide users with LBSs (e.g., [1, 4]) but there is a lack of a general and flexible framework that can be applied in different scenarios. To build such a general system by simply merging preexisting LBSs is not straightforward: it is a challenge to provide a common framework that allows 1) managing knowledge obtained from data sent by heterogeneous devices/users (textual data, multimedia data, sensor data, etc.); and 2) benefiting from direct P2P interactions among devices. In our opinion, the use of semantic techniques can enable the development of intelligent LBSs [3].

In this demo, we show *SHERLOCK*[2], a system based on the use of semantic techniques that relieves users from the need to obtain up-to-date information about the services they need. As its namesake, the well-known Arthur Conan Doyle's character, SHERLOCK uses abductive and deductive reasoning to infer information to answer user information requests (e.g., by providing interesting LBSs). Thus,

[1] *Google Play* returns approximately 78 results with the keywords "metro" and "Rio de Janeiro" and 38 results with "taxi" and "Rio de Janeiro" (February 2013).

[2] System for Heterogeneous mobilE Requests by Leveraging Ontological and Contextual Knowledge.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media.
WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

the system uses ontology reasoning and alignment methods to represent and manage, in a distributed way, the knowledge that describes moving and static objects (e.g., persons, vehicles, etc.) and interesting areas in a scenario. In this way, SHERLOCK's functionality is twofold: 1) the participating objects/devices cooperate and exchange data and knowledge (using fixed or ad hoc networks) to relieve the user from knowing and managing such knowledge directly; and 2) the system guides the user in the process of selecting the service that best fits his/her needs in the given context. New services can be easily incorporated into the system by defining in an ontology the background knowledge required, which will be shared and propagated through the interested mobile devices.

2. SYSTEM ARCHITECTURE

The basic idea behind SHERLOCK is a network of devices (mobile or static, wired or wireless, etc.) that share knowledge and offer services to each other. In this way, a device with SHERLOCK updates its knowledge when it interacts with other devices, learning from them, and so it relieves the user from the need to manage specific knowledge about his/her context. Moreover, in SHERLOCK's design we do not rely on a specific communication technology (the system uses both wired and ad hoc wireless P2P networks) or positioning technique for the devices, as the system adapts to the existing ones dynamically. The SHERLOCK architecture on a device has two main modules (see Figure 1):

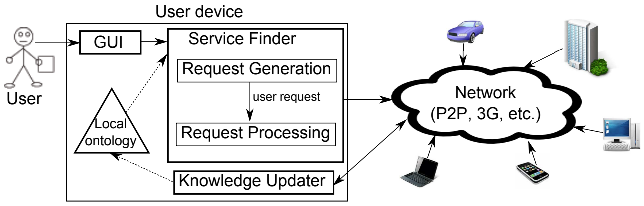


Figure 1: SHERLOCK's architecture on each node.

- *Knowledge Updater* (see Section 2.1): it shares and integrates into a local ontology [2] the knowledge that the system finds in other nodes. A special kind of knowledge that SHERLOCK manages is the definition of services.
- *Service Finder*: it is in charge of attending the user information needs. First, the *Request Generation* module (see Section 2.2) searches in the local ontology for the services that could be interesting for the user at that moment. Second, the user request is managed by the *Request Processing* module over the underlying cloud of (fixed and moving) objects in the scenario (see Section 2.3).

In the rest of this section we summarize the three main modules indicated.

2.1 Knowledge Updater

The Knowledge Updater is in charge of upgrading the local knowledge about objects, scenarios, services, etc., by interacting with Knowledge Updaters on other devices. Thus, the broader the knowledge SHERLOCK manages, the more useful the help it can provide to the user. When two SHERLOCK devices meet they share their knowledge, and the information exchanged is integrated into their local ontologies. In this way, the interaction with different devices upgrades the knowledge that the system manages, and so the information that it can offer to the user (e.g., it can show new services, improve the definition of a service with specific providers, etc). For example, a user that arrives in an area (e.g., a city) could automatically obtain information about that specific area as soon as the Knowledge Updater interacts with other SHERLOCK devices. Moreover, as the Knowledge Updater integrates ontologies and ontology fragments shared with the system, anyone able to describe knowledge can provide new knowledge for SHERLOCK

We are aware that privacy issues affect elements in the knowledge sharing process. For example, we could select to share a certain piece of knowledge with some specific objects or *profiles*, or we could prefer not to receive certain knowledge. To address this problem, cryptography for protecting sensible information and schemas based on digital signatures/certificates for authentication could be used by SHERLOCK, although this has not been incorporated yet to the demo described in this paper.

The Knowledge Updater performs its knowledge exchange task continuously and the new knowledge acquired benefits even services already being executed on the user device.

2.2 Request Generation

SHERLOCK provides the user with a service finder that helps him/her to define his/her information needs and obtain interesting information. In this case, as the system is focused on mobile users, the service finder is focused on LBSs (even though it could also process other types of services). First, the service finder deals with the generation of a user request by discovering his/her needs, using the information provided by the user as well as his/her contextual information (and, of course, the local ontology of the device). The system is guided by the definition of LBSs in the ontology, and so the following steps are followed whatever LBS (in Section 3 we show how this same process is performed for three different scenarios):

1. First, the user shows interest in a geographic area or object by typing some keywords in SHERLOCK's search bar (e.g., a user can input the name of a hotel or a city) or by clicking on it on a map. In addition, contextual information (i.e., the user's current location, time, etc.) is considered to enable SHERLOCK to deduce the most interesting LBSs for that user. For this task, SHERLOCK uses a DL reasoner [5] and the local ontology where the LBSs are defined. SHERLOCK considers an LBS as interesting if it is related (according to the knowledge represented in the ontology) to the information provided by the user[3].
2. Then, SHERLOCK presents the potentially interesting LBSs and requires the user to select one of them.

[3] SHERLOCK looks for services that are somehow related to the concept/instances selected by the user, whatever the name of the property referenced by such a concept (we do not assume any predefined schema in the definition of services).

These services could have different providers with specific features, so the user is asked for his/her preferences.

3. Finally, SHERLOCK takes the preferences of the user into account to obtain the most appropriate service. Moreover, the use of an ontology and a DL reasoner enables the system to detect potential situations where the user preferences cannot be fulfilled by any service provider, just by checking their ontological definitions.

Notice that the information provided by the user (a click on a map or keywords that define a location, selecting a service from a list, or filling a user-friendly form) is enough for SHERLOCK to retrieve interesting services that fulfill his/her demands for that geographic area and time. The next step is to process the user request to obtain the location and information of the objects that can provide the user with this service or that are related to it.

2.3 Request Processing

SHERLOCK processes user requests taking into account the information sources described in the local ontology. The matching services in the ontology will specify that the information requested by the user could be obtained from:

- Other objects in the scenario. SHERLOCK connects to other devices to retrieve the information interesting for the user request.

- Third-party information providers, such as web sites. For example, if an LBS has been defined in the local ontology making explicit that the information about its related objects can be obtained from the Web (e.g., using a Web service), then SHERLOCK will use it.

SHERLOCK shows information about the interesting objects obtained on a map, and continuously processes the user request (the details about how SHERLOCK communicates with other objects are beyond the scope of this demo). Indeed, this reevaluation is needed because the objects can move and the knowledge of SHERLOCK can be improved over time. Notice that SHERLOCK is able to process multiple user requests at the same time.

3. DEMONSTRATION

In the demo that we present as a motivating use case, we will use a laptop to edit an ontology and start distributing such knowledge to other devices. We focus on the case of a researcher attending the WWW 2013 conference, and show several situations where SHERLOCK could be useful. The SHERLOCK client for Android and some videos are available in our website (`http://sid.cps.unizar.es/SHERLOCK/Android/`). Concerning security and privacy issues, in this demo we trust all the shared knowledge and we enable the user to change his profile to test the system.

The user arrives in the Rio de Janeiro-Galeão International Airport and executes SHERLOCK. Then, the system starts searching for nearby available devices that have shareable knowledge. This process would enable SHERLOCK to learn local information (e.g., transportation means in the city, tourist information, etc.). This task will be executed in the background during the execution of SHERLOCK.

3.1 Looking for Transportation

The user wants to get to the conference hotel but does not know the best way to go there. He types its name "Windsor Barra Hotel" in SHERLOCK's search bar and a search using *Google Places API* provides SHERLOCK with his location (that is displayed on the map). Moreover, SHERLOCK finds the "Transportation Service" that would be interesting for the user, as he is searching a "Hotel" and this service has a parameter that is a "Destination" and "Hotel" is a possible destination (see Figure 2).

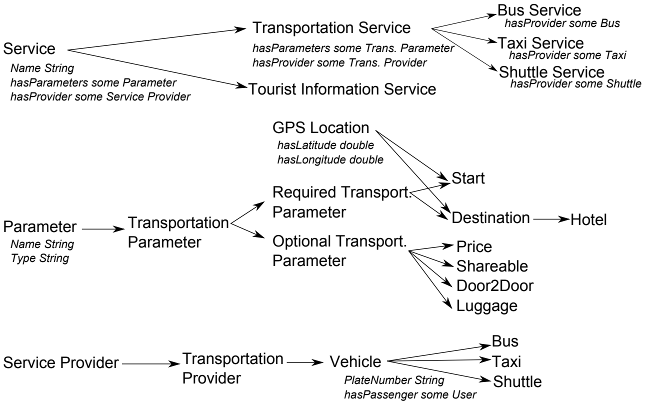


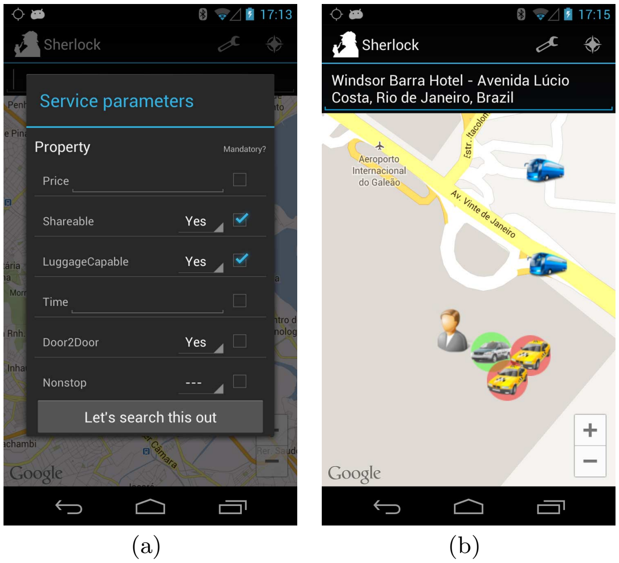
Figure 2: Subset of an ontology that defines an LBS: "Transportation Service".

The user selects the "Transportation Service" and by using the local ontology SHERLOCK reasons that it can be provided by different transportation vehicles, such as taxis, buses, and shuttles (knowledge about transportation means in the area has been provided by a device nearby). So, it presents to the user their common and specific features and allows the user to specify his preferences (see Figure 3(a)). He shows his interest in a transport *Shareable* that admits *Luggage* and *Door2Door*, if possible. Then, the system infers that *Shuttles* are the best option according to the knowledge that has been shared with it, but *Taxis* and *Buses* are also interesting. Figure 3(b) shows the interesting objects obtained.

3.2 Meeting Fellow Researchers

The conference organizers could have made some knowledge available in the hotel for customers registered in the conference (e.g., an ontology describing the attendees and defining a "Locate Fellow Researchers" service). The user is interested in locating fellow researchers from his same research field to talk to them, so the recently found service could be very handy. The ontological definition of the service establishes that it is available to researchers and the user has not selected this profile yet. However, the knowledge that the organizers shared described the user as an instance of the concept researcher along with information about his research area. In this way, SHERLOCK suggests the user to use the researcher profile.

Then, SHERLOCK will connect to devices in the neighborhood to locate those whose profile has been set to researcher and that share the same research area. When processing the request, SHERLOCK will ask other devices about their ontological definition, and so a researcher's device will share information about his research field. Thus, it could



Service parameters

Property Mandatory?

Price		□
Shareable	Yes	☑
LuggageCapable	Yes	☑
Time		□
Door2Door	Yes	□
Nonstop	---	□

Let's search this out

Windsor Barra Hotel - Avenida Lúcio Costa, Rio de Janeiro, Brazil

Figure 3: SHERLOCK Android app: (a) selection of parameters of a transportation service and (b) transports found

obtain information about research areas unknown to its local ontology. Finally, users are able to click on a researcher and SHERLOCK will present his personal information defined in the ontology (i.e., name and email), as well as other services related to a researcher, such as the "Appoint meeting by email" service. Figure 4(a) shows the information obtained when selecting a fellow researcher.

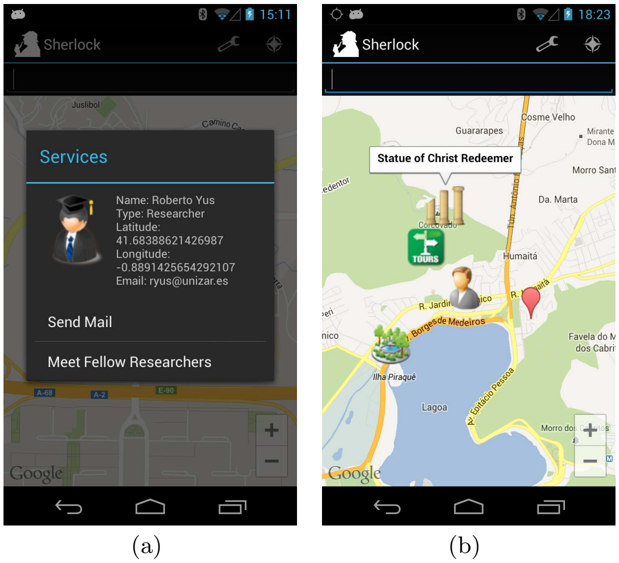
3.3 Tourist Information

After some productive meetings with his colleagues the user wants to sightsee Rio. Along with the knowledge of the area that SHERLOCK found at the airport, a tourist information stand could share a specific "Tourist Information" service with information about the points of interest of the city. So, when the user clicks on the city center on SHERLOCK's map, this service is offered. The ontological definition of the "Tourist Information" service relates it with different "static" objects, such as monuments, cathedrals, parks, etc. It also shows that there exists a related Web service that provides the location of some of these objects (Google Places). So, SHERLOCK will query the Web service and represent the information it obtains on the map.

Moreover, the service has been also related (in the ontology) with some moving objects, such as tourist buses, tourist guides, etc. So, SHERLOCK looks for these interesting objects thanks to the use of P2P communications with nearby devices. Finally, all the objects are shown on the map and the user can click on them to obtain further information. Figure 4(b) shows the monuments obtained from the Web service along with a tourist guide located nearby.

4. CONCLUSIONS

SHERLOCK is a system that searches and shares up-to-date knowledge between participating devices, which collaborate to relieve users from managing such knowledge directly. A special kind of knowledge that SHERLOCK manages is the definition of services, which is taken into account to provide users with the information they need at anytime.



Services

Name: Roberto Yus
Type: Researcher
Latitude:
41.68388621426987
Longitude:
-0.8891425654292107
Email: ryus@unizar.es

Send Mail

Meet Fellow Researchers

Figure 4: SHERLOCK Android app: (a) obtaining information about a researcher and (b) results obtained when searching for interesting touristic points

5. ACKNOWLEDGMENTS

This research work has been supported by the CICYT project TIN2010-21387-C02 and DGA-FSE. We would like to thank Guillermo Esteban for his help with the implementation of the Android application.

6. REFERENCES

[1] B. Gedik and L. Liu. MobiEyes: A distributed location monitoring service using moving location queries. *IEEE Trans. Mobile Comput.*, 5(10):1384–1402, 2006.

[2] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, Dec. 1995.

[3] S. Ilarri, A. lllarramendi, E. Mena, and A. Sheth. Semantics in location-based services – guest editors' introduction for special issue. *IEEE Internet Comput.*, 15(6):10–14, 2011.

[4] S. Ilarri, E. Mena, and A. Illarramendi. Location-dependent queries in mobile contexts: Distributed processing using mobile agents. *IEEE Trans. Mobile Comput.*, 5(8):1029–1043, 2006.

[5] R. B. Mishra and S. Kumar. Semantic web reasoners and languages. *Artificial Intelligence Review*, 35(4):339–368, 2011.

[6] J. Schiller and A. Voisard, editors. *Location-Based Services*. Morgan Kaufmann, San Francisco, CA, USA, 2004.