Continuous Processing of Real-Time Multimedia Requests Using Semantic Techniques

Roberto Yus
ryus@unizar.es

Eduardo Mena
emena@unizar.es

Dept. of Computer Science & Systems Engineering
University of Zaragoza
50018 Zaragoza, Spain

ABSTRACT

Mobile devices have penetrated our daily lives and currently smartphones and tablets are everywhere. These devices are equipped with different sensors which enable them to capture multimedia information (e.g., several cameras and microphone). Therefore, today is easier than ever to count with a device which could capture information of interest in real-time almost anywhere. We present an approach to enable users to specify the kind of real-time multimedia information they are interested in and to obtain such information (in a continuous manner) from remote devices based on: 1) semantic techniques to handle the knowledge associated with different scenarios and requests, and 2) a network of mobile agents to process these requests continuously over different devices and communication networks.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems – Query processing; H.4.0 [**Information Systems Applications**]: General

Keywords

Multimedia-based requests, location-based services, ontologies, mobile agents

1. INTRODUCTION

Accessing Google, Flickr, or Twitter may help to find multimedia information in some cases, but for many particular requests the only way will be to capture the requested information in real-time. Nowadays we could think of requesting devices equipped with cameras located near an interesting place to obtain and share pictures/videos. Security and trust issues arise in this case but the great possibilities of such a collaborative decentralized information system could be worthy for certain scenarios such as emergency management or crime prevention/solving, as well as many others like becoming a voluntary provider of real-time multimedia data for news or sport events broadcasting, among others.

In this paper we present how the SHERLOCK system [10], which provides users with interesting Location-Based Services (LBSs), has been extended to manage user requests to obtain real-time multimedia information from certain interesting geographic areas. SHERLOCK helps users to specify their information needs and manages LBSs defined using OWL ontologies, and also it is in charge of the communication among the user and those devices where interesting data are (or from which data can be obtained). Here we describe how the SHERLOCK architecture handles multimedia information and the challenges associated with it such as enabling the user to specify the (multimedia) features of the information requested, locating possible multimedia information providers (including requesting users to capture certain multimedia information needed), and processing (in real-time) camera views to check if they fulfill the requirements of a user, performing all the previous tasks in a continuous and efficient way. In summary, the main features of the system presented in this work are:

- It is based on ontologies that support diverse LBSs to obtain multimedia information. Thus, a single system is able to serve for many different purposes.
- It finds and communicates with devices which have or could capture the multimedia information requested by the user.
- It can even interact with other users to ask them to capture multimedia information, taking into account their privacy and trust policies.

The rest of the paper is as follows. In Section 2 we present two motivating scenarios and their challenges In Section 3 we summarize the main architecture of SHERLOCK. In Section 4 and Section 5 we describe how a multimedia request is built and processed, respectively. Related works are presented in Section 6 and, finally, conclusions and future work are presented in Section 7.

2. MOTIVATING SCENARIOS

There are plenty of situations where users might be interested in obtaining real-time multimedia information from devices in an area. In the following, as an example, we show two different motivating scenarios and the challenges that a system to handle them would face.

MoMM 2015, December 11-13, 2015, Brussels, Belgium
© 2015 ACM. ISBN 978-1-4503-3493-8/15/12…$15.00
DOI: http://dx.doi.org/10.1145/2837126.2837170

2.1 Live Broadcasting of Sport Events

The Technical Director (TD) in charge of a live broadcasting has to make quick and frequent decisions to select the camera to broadcast. In addition to the increasing number of cameras used by broadcasting organizations, the audience of the sport event could provide TDs with interesting shots too. For example, consider the TD in charge of the live broadcasting of a rowing race in San Sebastian (Spain). From the TD perspective, the scenario includes multiple cameras available for the broadcasting (in the rowing boats, in a helicopter, in the harbor, and in a nearby island) and many people among the crowded audience of the event who are equipped with mobile devices (e.g., smartphones and tablets). In this context, it would be very helpful to have a system where the TD could define her interest on a certain view (e.g., *a view of the front of two rowing boats*) and obtain the list of cameras (including broadcaster and audience cameras) that could provide it, currently or in a matter of seconds.

2.2 Emergency Management

Many well-known real emergency situations have shown that without a good coordination and information flow, the task of emergency teams is very difficult, sometimes becoming very dangerous for themselves. The bigger the scale of the emergency, the more valuable any information about the real situation; and multimedia information can become priceless in almost 100% of emergencies, especially when such data are obtained in real-time and propagated to emergency teams. Let us imagine that an accident has occurred in the highway. The emergency team that received the alert needs to know the location of emergency vehicles and personal (e.g., policemen, firemen, etc.) in the surroundings which could help. Real-time multimedia information (images/videos) of the accident area would be very useful for the emergency team to determine the severity of the accident and drivers in the vicinity of the accident would be willing to help because lives could be in danger.

2.3 Challenges

Some common challenges arise in the above scenarios for a system to manage them. So, the system should be able to: 1) provide a flexible user interface able to help the user to define the kind of multimedia information needed (his is specially challenging for the scenario of TV broadcasting); 2) track the location of objects of interest (the rowing boats in the first scenario and objects and emergency teams around the accident area in the second) as well as the cameras (owned by the broadcaster in the first scenario and from other users in both scenarios); 3) discover users equipped with cameras and ask them to share pictures/videos they took recently or even to capture more (following privacy and trust policies is mandatory in this case); 4) analyze the multimedia information retrieved and the features of the devices in real-time to discard those users/devices that could not be able to provide the requested multimedia data and to rank the results obtained by the system according to the matching with the kind of answer wanted; and 5) maintain a continuous communication between the user and the rest of the devices involved in the distributed scenario in order to provide the final user with an updated answer continuously.

3. ARCHITECTURE OF THE SYSTEM

In this paper we present an extension of the SHERLOCK architecture [10], a general system to provide Location-Based Services (LBSs) to mobile users. SHERLOCK is based on: 1) ontologies and semantic reasoners based on Description Logics (DL) to handle the knowledge about LBSs and interesting objects and events, and 2) mobile agents to balance the CPU consumption and communication load. We here show how the system is able to handle multimedia information and to request users to obtain information for others. SHERLOCK-enabled devices are able to communicate and cooperate among them. The information that SHERLOCK obtains is extracted from the local sensors on the device where it executes or obtained from remote SHERLOCK devices. Also each SHERLOCK device manages its local ontology (the knowledge of that user device) without interactions with the user. For more details see [10].

In the following we summarize the two main steps followed by SHERLOCK to accept and process location queries involving multimedia information:

1. *Creating User Request*: This step includes the selection of the LBS that the user needs and the capture of the specific user information needs through interfaces designed to handle the definition of the multimedia information needed.

2. *Processing User Request*: This step includes the automatic generation of a formal query which models the user request and its processing against the local knowledge on the device, third party repositories, or even other devices.

In the following sections we explain these steps in detail.

4. CREATING A USER REQUEST

To define a multimedia request, the system assists the user in the selection of an appropriate service and captures her needs as follows.

4.1 Selecting Appropriate Services

The information about services and scenarios that the system manages is modeled as ontologies [3]. This way, new scenarios and services can be defined by providing similar ontologies and the system will be able to handle them. We selected the OWL language to encode the ontologies as its expressivity enables the definition of complex knowledge, if needed. Also, to infer information that has not been explicitly defined we use a semantic reasoner based on Description Logics. SHERLOCK handles a structure of services, from which the user can select, based on some predefined concepts and roles which other services can extend such as: "Service", "Provider", "Parameter", and "Result". As an example of the definition of services and knowledge related to them, in the following we present the knowledge that the system needs to handle the motivating scenarios.

Common Knowledge. The two motivating scenarios described before (and many others) need a service to obtain pictures and videos. We defined the general services "GetPhotos" and "GetVideos" (see Figure 1 for an excerpt of the ontology. These services have as providers entities equipped with a camera ("hasCamera some Camera"), as results photos and videos, respectively, and parameters such as the distance from the camera to objects inside and the visibility of such objects in the field-of-view of the camera.

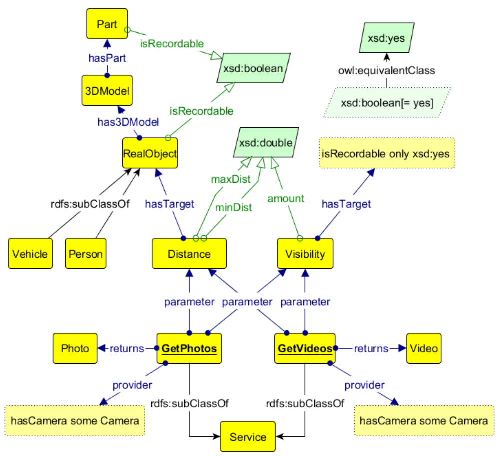

Figure 1: Services "GetPhotos" and "GetVideos".

Specific Knowledge for the First Scenario. We have defined a service to manage the broadcasting of a sport event (see Figure 2 for an excerpt of the ontology) and a specific instance of this service for the rowing race in the example ("BroadcastingBanderaConcha2015"). Notice that, this service is composed of two services to obtain information about the rowing boats participating in the race and the cameras of the broadcaster. Also, we have defined the knowledge related to the rowing race which is used in combination with the previous definition of the service (e.g., notice that the service to find the rowing boats has as result the rowing boats participating in the race). Among the information modeled is the participants of the rowing race and cameras managed by the broadcasting company, as well as possible interesting locations for the broadcast (such as the "ciaboga" area which is the turning point for the boats). All this knowledge would be defined by the broadcasting company and its information system could share it as an ontology with the SHERLOCK on the TD device.

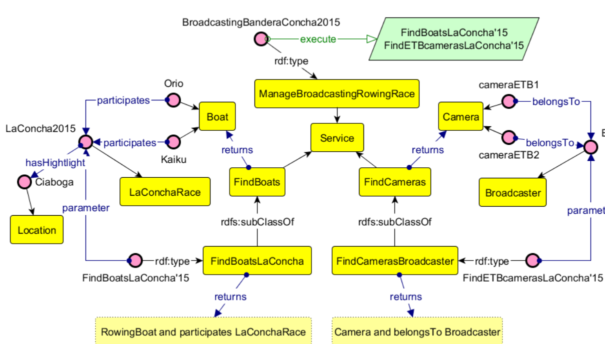

Figure 2: Knowledge about a rowing race.

Specific Knowledge for the Second Scenario. For the second use case we have defined a service to manage emergencies (see Figure 3 for an excerpt of the ontology). This service gets information from the accident ("FindAffectedVehicles"), finds emergency teams in an area defined by the user ("FindEmergencyTeams"), then assign these teams to the accident ("SendTeam"), obtains pictures and videos from the accident area ("GetPhotos"), and finally sends the multimedia content to emergency units ("SendPicture"). As in the previous scenario, this service would be defined by an authority and shared with the SHERLOCK on the coordinator device that will integrate it into its local ontology.

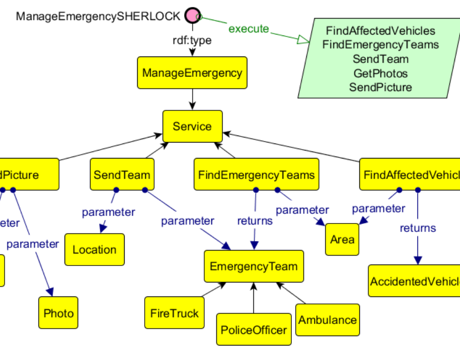

Figure 3: The "Manage Emergency" service.

4.2 Capturing the User Needs

After the user selects a service the system asks her to fill in its parameters. The system handles simple parameters such as text and booleans but also locations through a map interface (e.g., a Google Maps map). For example, the TD of the first motivating use case might want to select the "ciaboga" area to find boats around it. Also, the system offers a mechanism to define specific images/videos the user is interested in. This mechanism is useful in situations where the user has an idea about (or even knows exactly) the kind of image she wants or if plenty of images are available and they should be ranked according to a criteria more complicated than the actual distance to the target location. Our system uses an interface based on the 3D Query-by-Example (3DQBE) interface that we presented in [8] adapted to mobile devices. On that interface, the user defines the kind of shot to obtain by rotating the camera view and even including other objects in the scene (which the interface translates into a description such as "an image showing 30% of the front view and 50% of the top view of the Kaiku boat").

5. PROCESSING A USER REQUEST

To process a user request the system first formalizes it into a query and then executes it against different sources.

5.1 Generating the Formal Query

In order to make SHERLOCK very flexible and useful for many scenarios, the user request should be (automatically) expressed as a query in a formal language; and the system should be able to process any query in that language to obtain its corresponding answer. Thus the system becomes usable, not only for humans using a GUI, but also for third party systems posing queries in such a language. In our system, we propose a language based on SPARQL, a query language able to handle RDF data, as the underlying language to express the request. As we are dealing with requests that are related to locations and we use OWL ontologies, we use the GeoSPARQL and SPARQL-DL extensions of SPARQL (see [9] for more information about SHERLOCK's language). The information captured in the previous step is translated into a query in SHERLOCK's language for its later processing.

5.2 Query Processing

To process a query, SHERLOCK tries to get information from different sources (see the high-level flow diagram of Figure 4). First, SHERLOCK executes the query against the local ontology on the device: It is possible that the system already has the information requested (e.g., in our case the multimedia information requested and/or information about devices which have or can capture the multimedia information that the user wants) because of the processing of previous queries. Notice that if results are obtained from the local ontology SHERLOCK checks the number of results and their timestamp to make sure they are valid for the user. In the case of not obtaining enough results or the results being out-of-date, SHERLOCK tries to obtain the information from other sources. First, it tries to obtain it from third party external resources defined in the SHERLOCK's ontology (e.g., DBpedia or Flickr) that could have the needed information.

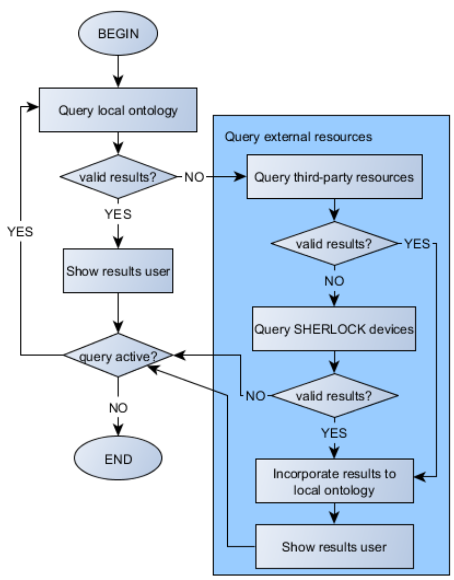

Figure 4: Flow of execution of a multimedia query.

In some situations querying third-party resources could return an empty or (more frequently) out-of-date result. Then, SHERLOCK tries to obtain information directly from devices in the area of interest. For this task, SHERLOCK creates a network of mobile agents that move near the interesting area in order to discover devices there and query them directly. The specifics of creation and maintenance of such a mobile agent network are out of the scope of this paper (see [10] for some details). Nevertheless, in the following we explain the most important steps, focusing on their multimedia aspect, to describe the high-level protocol involved:

1. Split the query for each non-overlapping geographic constraint. There will be a Tracker agent in charge of monitoring the geographic area associated with each geographic constraint. In the case of the first scenario, the area of interest is "La Concha" (San Sebastian bay). In the case of the second scenario, the area of interest is defined around the GPS coordinate of the accident.

2. Send agents to devices in the relevant area. Each Tracker agent creates mobile agents that will move to devices from which obtain information about the whole relevant area that such a tracker agent has to monitor. These agents will execute the query against such devices. For example, in the first scenario the agents in charge of tracking the rowing boats will try to get as close as possible to them moving to boats around or even to the rowing boats if needed. In the second scenario, the agents monitoring the accident will move (may be in different hops) to devices on the different vehicles or other entities (people, buildings) around the accident.

3. Each device executes the query against its local ontology and returns as answer those (multimedia) data fulfilling the query constraints that devices have already captured; e.g., stored videos of the rowing boats in the first scenario and locally stored images of the accident in the second scenario (if any). In case that those devices do not have captured the requested multimedia information, the agents obtain information about their cameras. With this information it is possible to determine whether they could capture the requested information or not as follows:

- To evaluate if a device can capture a requested photo or video, SHERLOCK incorporates the approach presented in [11] as a module. That approach considers information from the devices (and their cameras) such as their location, direction, and field-of-view as well as information from the target of the user request such as its location, direction, and an approximation of the volume which it occupies (e.g., a 3D model of the object or a simple box. With that information, the system is able to estimate if a camera is viewing the target object (a rowing boat, a car, a museum, etc.) and it also estimates the kind of image it would take. The module also provides SHERLOCK with information about the rotation that a camera should perform to view the target. In some cases human intervention can be needed to capture the information.

- To ask a user to obtain multimedia information the agent communicating with her SHERLOCK sends a petition to it (which includes the maximum amount of time the agent is willing to wait for a reply). Then SHERLOCK on her device checks the context of the user (which might show that she is busy) and her preferences regarding requests from others and asks the user for collaboration. It communicates the agent if she accepts and in that case, an agent will be sent to the user device in order to manage such a request (e.g., to show the rotation of the camera needed to capture the target video, or where to point at before taking a picture) and then return the result.

Finally, the results obtained by each agent are correlated (selecting the most updated data in case of duplicates) and sent to the user device which presents them on the user device. The process of obtaining multimedia information from the different sources is performed continuously until the user cancel her request. During the whole query processing, mobile agents are continually located on the best device to achieve their goals, jumping from one device to another wherever needed or it is considered a better place to do their tasks.

SHERLOCK takes into account the privacy issues that arise in these scenarios by considering user preferences when accessing their devices. For that, SHERLOCK adopts the semantic approach to define privacy preferences proposed in [1] which model them using the Semantic Web Rule Language (SWRL). Using SWRL the user can define, for example, that friends can get multimedia information captured by her device after asking her or that certified authorities can get her multimedia information without notifying the user. These SWRL privacy policies are automatically evaluated on the user device by SHERLOCK using the semantic reasoner whenever a request for information is received.

6. RELATED WORK

To our knowledge, our system is the first single flexible system that provides users with a general method to request and obtain real-time multimedia information from anywhere in a distributed manner. Therefore, we describe in this section works focused on our two motivating use cases.

Concerning the broadcasting of sport events scenario, the CrowdDirector system [6] enables users to capture videos and upload them to a server where the composition tasks is outsourced to workers. The MediaQ system [4] enables users to define their media content needs as tasks which are sent to a server to select workers in the area that can fulfill it. Finally, the MultiCAMBA system (Multi-CAMera Broadcasting Assistant) [11] helps TDs by enabling them to indicate in run-time their interest in certain kind of shots and showing the cameras that are able to provide them. As opposed to these systems, SHERLOCK is a distributed architecture where the sources of information do not have to register or upload their content to an external server and can be found by the system.

Regarding the emergency management scenario, the interest of real-time user generated content for emergency and disaster response has been highlighted in [2]. In the specific scenario of traffic accidents the use of communications between vehicles has been exploited for video streaming [5]. However, these works are focused on how to efficiently distribute the content in VANETs whereas we are focused on the complete process from defining the information needed to finding the devices which are providing (or could provide) the multimedia information requested. In [7] a system is presented to enable users to upload geo-annotated pictures to a server where emergency coordinators can access the information. In their centralized approach users have to proactively upload the content, whereas SHERLOCK finds the users and obtains the information automatically, whenever possible.

In summary, despite these systems work well for the task they were designed for, they are not general multimedia query processors. For example, these systems cannot solve by themselves the two scenario described in this paper neither other similar use cases, without modifications.

7. CONCLUSIONS

In this paper we have presented a system which enables users interested in obtaining multimedia information to request it to other users and/or devices which can capture it. Our approach enables users to specify the kind of multimedia information they want to obtain; for this, the system is based on semantic technologies to handle the knowledge associated with different scenarios and requests. It also obtains and accesses the remote devices that could capture the multimedia information requested, and returning the answer to the user in a continuous manner; for this task it spreads a network of mobile agents that in real-time keep computation wherever needed to efficiently select the data that fulfill the user requirements. Although focused on the main architecture of the system, we reuse and adapt techniques from related work in order to semantically describe multimedia services as well as to analyze multimedia data captured. As examples of requests of multimedia information we have presented the efficient processing of pictures and videos in two different scenarios (TV broadcasting and emergency management) that can be solved by a single system.

As future work, we plan to consider the specific features of audio information and how to process it efficiently on mobile devices. Also, we plan to finish a complete prototype of the system to perform experiments with real users.

Acknowledgments. This research work has been supported by CICYT project TIN2013-46238-C4-4-R and DGA-FSE. We want to thank Carlos Bobed for his insightful comments.

8. REFERENCES

[1] D. Beimel and M. Peleg. Using OWL and SWRL to represent and reason with situation-based access control policies. *Data Knowl. Eng.*, 70(6):596–615, 2011.

[2] M. F. Goodchild and J. A. Glennon. Crowdsourcing geographic information for disaster response: a research frontier. *Int. J. Digital Earth*, 3(3):231–241, 2010.

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

[4] S. H. Kim, Y. Lu, G. Constantinou, C. Shahabi, G. Wang, and R. Zimmermann. MediaQ: Mobile multimedia management system. In *Fifth ACM Multimedia Systems Conf. (MMSys 2014)*, pages 224–235, 2014.

[5] J.-S. Park, U. Lee, S. Y. Oh, M. Gerla, and D. S. Lun. Emergency related video streaming in VANET using network coding. In *Third Int. Workshop on Vehicular Ad Hoc Networks (VANET 2006)*, pages 102–103, 2006.

[6] S. Wilk, S. Kopf, and W. Effelsberg. Video composition by the crowd: A system to compose user-generated videos in near real-time. In *6th ACM Multimedia Systems Conf. (MMSys 2015)*, pages 13–24, 2015.

[7] A. Wu, X. Yan, and X. L. Zhang. Geo-tagged mobile photo sharing in collaborative emergency management. In *2011 Visual Information Communication - International Symposium (VINCI 2011)*, pages 7:1–7:8, 2011.

[8] R. Yus, S. Ilarri, and E. Mena. Real-time selection of video streams for live TV broadcasting based on query-by-example using a 3D model. *Multimedia Tools and Applications*, 74(8):2659–2685, 2015.

[9] R. Yus and E. Mena. Mobile endpoints: Accessing to dynamic information from mobile devices. In *14th Int. Semantic Web Conf. (ISWC 2015)*, 2015.

[10] R. Yus, E. Mena, S. Ilarri, and A. Illarramendi. SHERLOCK: Semantic management of location-based services in wireless environments. *Pervasive and Mobile Comp.*, 15(0):87–99, 2014.

[11] R. Yus, E. Mena, S. Ilarri, A. Illarramendi, and J. Bernad. MultiCAMBA: A system for selecting camera views in live broadcasting of sport events using a dynamic 3D model. *Multimedia Tools and Applications*, 74(11):4059–4090, 2015.