Building SPARQL-Enabled Applications with Android Devices*

Mathieu d'Aquin, Andriy Nikolov, Enrico Motta

Knowledge Media Institute, The Open University, Milton Keynes, UK
{m.daquin, a.nikolov, e.motta}@open.ac.uk

Abstract. In this paper, we show how features can be added to an Android device (a smartphone) to enable mobile applications to expose data through a SPARQL endpoint. Using simple query federation mechanisms, we describe a demonstrator illustrating how SPARQL-Enabled Android devices can allow us to rapidly develop applications mashing-up data from a collaborative network of sensor-based data sources.

1 Introduction

In recent years, the Android platform (http://www.android.com/) became a defacto standard for different types of mobile devices from several manufacturers. These devices possess several types of embedded sensors such as a camera, an accelerometer, a GPS sensor and a microphone. On the other hand, as shown by our previous work [1], the computational power of these devices already allows efficient processing of small to medium volumes of semantic data.

In this paper, we describe a lightweight architecture that adapts and deploys a triple store on an Android device, providing a shared, local repository for Android-based applications to populate. The information gathered through this shared repository is exposed using an externally accessible SPARQL endpoint, making it possible to build applications that exploit data collected from a network of devices through query federation.

2 Overview

The idea on which this paper is based on is very simple: making data created from applications and sensors attached to an Android device exposed through a SPARQL endpoint so that this data can be gathered by application, potentially aggregating multiple devices (see Figure 1).

The idea is that application developed for the Android platform can produce data to be stored in the shared repository on the device, so that it can be made accessible, without the need for post-processing, to external applications.

3 Implementation

At the core of our approach is the deployment of a triple store on the Android device, which is shared by applications populating it and by the SPARQL endpoint deployed in the device. As discussed, in [1], Sesame (http://www.openrdf.org/) is, amongst the available options, the one that best fits an environment where

* Part of this research has been funded under the EC 7th Framework Programme, in the context of the SmartProducts project (231204).

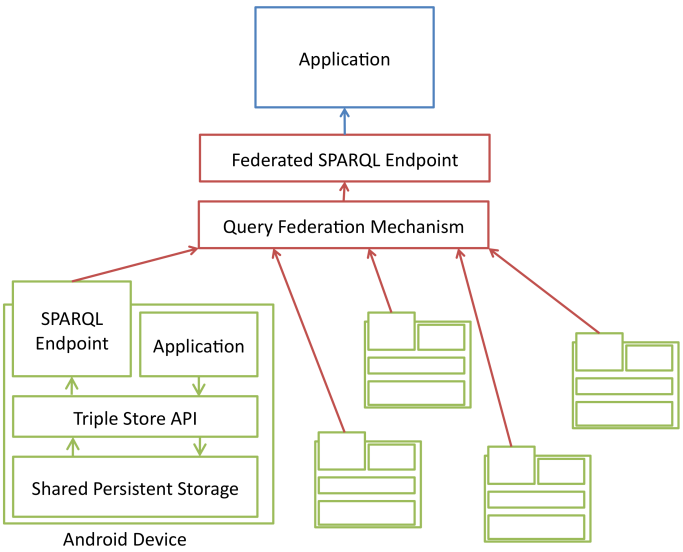

Fig. 1. Overview of the approach to create Semantic Sensor Networks out of Android devices.

only limited resources are available. We therefore adapted Sesame to be deployable as an Android Library. The Android environment is based on a specialised Java Virtual Machine, and Sesame being developed entirely in Java, most components of its components did not require any adaptation. Access to files is however different on the Android platform than it is on a usual computer. We therefore extended Sesame so that it provides a persistent RDF store using the shared, external storage available on most Android Devices (in SmartPhones, it corresponds to the SDCard). In other terms, a shared persistent repository is installed on the Android device that is accessible, and can be populated, by applications using the device's sensors.

The other element to be included on the Android device is a Web interface giving access, through the SPARQL protocol, to the content of the shared triple store. One of the difficulty here is to deploy a Web server on the phone, being accessible externally. Luckily, the popular Web application server Jetty (http://jetty.codehaus.org/jetty/) has been ported to work on the Android Platform in iJetty (http://code.google.com/p/i-jetty/), providing both a Web server and a servlet environment. We therefore implement (a simplified version of) a SPARQL endpoint as a web application relying on our Android-adapted version of the Sesame API.

Finally, a mechanism is needed for the federation of SPARQL query over the various Android devices. In the cases where the data comes only from isolated and independent sources, this federation mechanism can be very simple, as it only requires concatenating the results obtained from different devices. In more complex scenarios where information from different sensors can be linked, a more sophisticated mechanism is needed. We rely here on our own implementation of

a distributed SPARQL query endpoint based on federating queries to multiple other SPARQL endpoints (see [2]).

The base code and components to deploy and populate a SPARQL endpoint on an Android device are available at http://code.google.com/p/android-sparql/ (see in particular the basic documentation at http://code.google.com/p/android-sparql/wiki/Deploying_a_SPARQL_Endpoint_and_Populating_the_triple_store).

4 Example Application

To illustrate the benefits of the architecture we are proposing, we developed a simple application used to collaboratively "map" a geographical area using pictures (for example, to give an idea of the views at certain points of a path, in an area where Google Streetview does not cover, such as a University Campus).

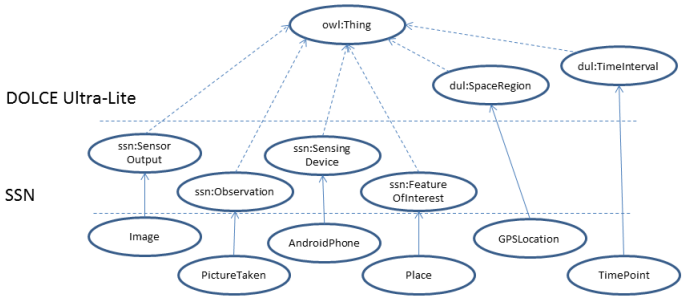

Fig. 2. Extension of the SSN ontology used in our example application.

The Application can take pictures and represent the information about the picture and its location as an Observation using the extension of the Semantic Sensor Network ontology shown in Figure 2. This application records the path of the picture on the device, the location of the device at the time of taking it, as well as the time and the identifier of the device used to make the observation. We used this application with several different SmartPhones. Using the simple SPARQL federation method described above, we implemented a Javascript application that displays the pictures taken from this network of phones into a map of the covered area (see Figure 3).

5 Related Work

At the moment the range of semantic data management tools specially targeted for resource-constrained devices is limited. MobileRDF[1] and microJena[2] (or μJena) provide Java APIs to load and manipulate RDF data on a device running Java ME virtual machine. Neither MobileRDF nor μJena allow querying these data using the SPARQL query language: only Java API access is provided. μOR [3] represents an ontological reasoner optimised for resource-constrained devices. μOR supports a subset of OWL-Lite axioms for reasoning and provides

[1] http://www.hedenus.de/rdf/
[2] http://poseidon.elet.polimi.it/ca/?page_id=59

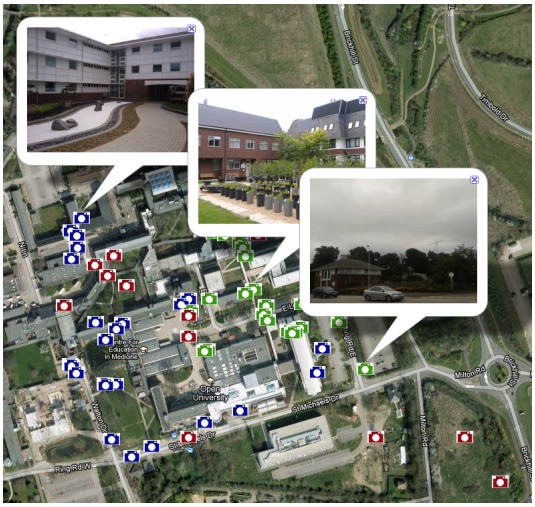

Fig. 3. Application mapping pictures from Android phones with SPARQL endpoints.

its own query language (SCENT - Semantic Device Language for N-Triples), which implements a subset of SPARQL's expressivity, to access data. However, none of the tools mentioned above provides a triple store for permanent storage of semantic data. One existing experimental storage solution is i-MoCo [4] which provides an experimental implementation of a triple store for the iPhone platform. For the Android platform, Androjena[3] adapts the well-known Jena framework[4]. However, this solution only allows working with a local store without the possibility to access distributed data. Mobile, Android applications such as [5] exist that exploit RDF and SPARQL, but mostly as clients of external, server side SPARQL endpoint.

References

1. d'Aquin, M., Nikolov, A., Motta, E.: How much semantic data on small devices? In: EKAW 2010, Conference - Knowledge Engineering and Knowledge Management by the Masses, Lisbon, Portugal (2010) 565–575
2. Miche, M., Erlenbusch, V., Allocca, C., Nikolov, A., Mascolo, J.E., Golenzer, J.: Final concept for storing, distributing, and maintaining proactive knowledge securely. Technical Report D4.1.3, SmartProducts Consortium (2011)
3. Ali, S., Kiefer, S.: muOR - A Micro OWL DL Reasoner for Ambient Intelligent Devices. In: 4th International Conference on Advances in Grid and Pervasive Computing (GPC 2009), Geneva, Switzerland (2009) 305–316
4. Weiss, C., Bernstein, A., Boccuzzo, S.: i-MoCo: Mobile conference guide – storing and querying huge amounts of Semantic Web data on the iPhone/iPod Touch. In: Billion Triple Challenge 2008, ISWC 2008, Karlsruhe, Germany (2008)
5. d'Aquin, M., Zablith, F., Motta, E.: wayou – linked data-based social location tracking in a large, distributed organisation. In: Proc. of the Extended Semantic Web Conference, ESWC (demo). (2011)

[3] http://code.google.com/p/androjena/
[4] http://jena.sourceforge.net/