Developing Mobile Linked Data Applications

Oshani Seneviratne[1], Evan W. Patton[2,1], Daniela Miao[1], Fuming Shih[1], Weihua Li[1], Lalana Kagal[1], and Carlos Castillo[3]

[1] Massachusetts Institute of Technology
{oshanis,ewpatton,dmiao,fuming,wli17,lkagal}@csail.mit.edu
[2] Rensselaer Polytechnic Institute pattoe@rpi.edu
[3] Qatar Computing Research Institute chato@acm.org

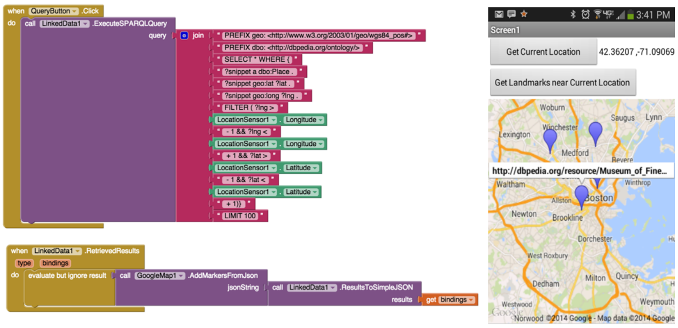
Abstract. With the rapid advancement of mobile technologies, users are generating and consuming a significant amount of data on their handheld devices. However, the lack of Linked Data tools for these devices has left much of the data unstructured and difficult to reuse and integrate with other datasets. We will demonstrate an application development framework that enables the easy development of mobile apps that generate and consume Linked Data. We also provide a set of easy-to-deploy Web Services to supplement functionality for mobile apps focused on crowdsourcing. We motivate our work by describing a real-world application of this framework, which is a disaster relief application that streams crowd-sourced reports in real time.

1 Introduction

Many developers are shifting their attention to the mobile world as smartphones are becoming the information hub for people's daily lives [1]. The pervasiveness of smartphones has led to the ubiquitous consumption and generation of data on them. Smartphones can derive contextual information from their environment, enabling applications that provide great value both to individual users and to society. For example, context-aware applications can recommend products, services, or connections to others based on people's surroundings. People can also use social applications to report their status and the situation around them in cases of emergencies, such as floods or earthquakes.

Most mobile applications create or consume data stored in standalone databases without the potential of being "interlinked" with data from other applications or organizations. The Web community has advocated the use of Linked Data technologies to address this data interoperability issue in Web-based applications. Although we have seen some success from content publishers in using or publishing Linked Data, few examples exist for mobile platforms [2].

Our goal is to bring support for Linked Data to mobile platforms by allowing developers to build mobile applications that consume and publish Linked Data automatically. We will demonstrate a framework that allows developers to accomplish this on Android devices through modular components, as well as cloud scripts aimed at enabling quick deployment of companion web services to interact with streams of Linked Data. We believe that this framework will reduce the burdens of mobile developers to work with Linked Data and open up many opportunities for building applications that interact with and contribute to the Linked Data world.



```
when QueryButton .Click
do  call LinkedData1 .ExecuteSPARQLQuery
                                   query  join  " PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> "
                                                " PREFIX dbo: <http://dbpedia.org/ontology/> "
                                                " SELECT * WHERE { "
                                                " ?snippet a dbo:Place . "
                                                " ?snippet geo:lat ?lat . "
                                                " ?snippet geo:long ?lng . "
                                                " FILTER ( ?lng > "
                                                LocationSensor1 . Longitude
                                                " - 1 && ?lng < "
                                                LocationSensor1 . Longitude
                                                " + 1 && ?lat > "
                                                LocationSensor1 . Latitude
                                                " - 1 && ?lat < "
                                                LocationSensor1 . Latitude
                                                " + 1)} "
                                                " LIMIT 100 "

when LinkedData1 .RetrievedResults
 type  bindings
do  evaluate but ignore result  call GoogleMap1 .AddMarkersFromJson
                                          jsonString  call LinkedData1 .ResultsToSimpleJSON
                                                                         results  get bindings
```

Fig. 1: **Consuming Linked Data.** *Left:* Logic used for obtaining the user's location, constructing a SPARQL query, querying a remote endpoint, and displaying the retrieved results on a map. *Right:* Output as seen on the phone. 'QueryButton' in the blocks editor is the name of the button with the label 'Get Landmarks near Current Location' as seen on the phone.

2 Framework

Our framework[1] extends MIT App Inventor,[2] an open source, web-based environment for developing Android applications. Though primarily designed for pedagogical use, App Inventor has found great success in many domains and boasts about 87,000 users weekly, 1.9 million total users, and over 4.9 million applications ranging from tracking rainfall in Haiti[3] to teaching U.S. Marines how to destroy weapons.[4] App Inventor is structured around *components*: discrete blocks that provide functionality and user interface elements that developers connect together to create an application. App Inventor provides a designer view for laying out an application's user interface and a blocks view where users define their program logic by connecting method blocks, associated with components, together (as shown in Fig. 1).

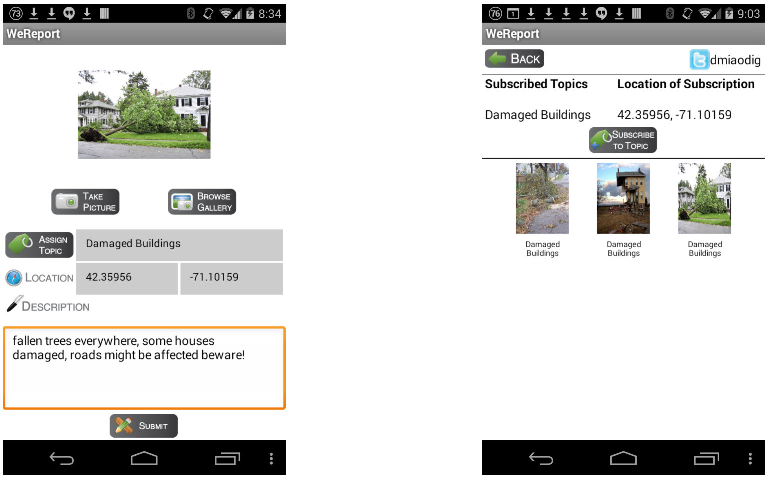
We will demonstrate several components that we've developed powered by Apache Jena[5], including forms to generate Linked Data, maps for visualizing geographic data, web services for cloud messaging, web services for exploring Linked Data, and sensor components for context awareness. We will also demonstrate companion web services that provide cloud functionality for mobile phones such as providing messaging services to integrate streaming RDF content.

3 Use Case: WeReport

WeReport allows people to report (e.g. through a photo) the situation on the ground during an emergency, hence enhancing situational awareness with respect to the emergency [3]. The application demonstrates the capacity to generate and consume Linked Data, as well as integrate with different public datasets.

Consider a scenario where a hurricane has hit a city, and Joe, a volunteer citizen reporter, notices a series of potential hazards in his neighborhood, e.g. fallen trees blocking an intersection. With WeReport, Joe can take a picture of the hazard and upload it, along with a tag and description, to warn others in the area. An example of this report can be seen in Figure 2a.

[1] http://punya.appinventor.mit.edu
[2] http://appinventor.mit.edu/
[3] http://www.webcitation.org/6PUXWVG0U
[4] http://www.webcitation.org/6PUXZb7FM
[5] https://jena.apache.org



Damaged Buildings

42.35956 | -71.10159

fallen trees everywhere, some houses damaged, roads might be affected beware!

Subscribed Topics | Location of Subscription

Damaged Buildings | 42.35956, -71.10159

(a) Submitting Reports (b) Browsing Reports

Fig. 2: WeReport application for Mobile Devices: an application that allows users to submit disaster reports to the cloud. Users can subscribe to reports close to a certain location on a topic such as 'Fallen Trees', and results are streamed in real-time to subscribers via push notifications.

When creating a report, WeReport consumes Linked Data via pre-defined SPARQL queries that search for popular landmarks near Joe's location, so that those places can be tagged if hazardous. At the same time, the mobile application enables him to transparently produce structured data through our Linked Data-enhanced components. The generated data can be published to SPARQL 1.1 compliant endpoints, which enables data re-use by others.

WeReport also supports real-time streaming of user-submitted disaster reports. In this scenario, Bob and Anna are both relief workers heading towards the affected area. Using WeReport, they have subscribed to the hurricane disaster feed and they continuously receive push notifications as reports on the disaster area are submitted by users like Joe. While Bob wishes to be continuously notified on every newly submitted report, Anna chooses to receive an alert only when there are at least 3 reports of a certain type in a given location, a custom limit she had previously specified. Whenever Bob and Anna want to view the newest reports, WeReport provides a "Browse Reports" interface which automatically displays the received reports, as show in Figure 2b.

4 Related Work

There has been some work in Linked Data use for smartphones, however, they do not provide a comprehensive solution to mobile app development as our framework does. [4] propose a lightweight general-purpose RDF framework for Android to share application data inside the phone. Our framework focuses on obtaining external RDF sources and consuming them and not necessarily application data integration using RDF. *Spinel* [5] is a plug-in architecture and a set of web configuration tools for Android that enable new datasets to be added to mobile phones without programming. We provide both a plug-in architecture and support for adding new datasets using program blocks that can be configured easily. Another notable platform for creating mobile Linked Data applications is the "RDF on the go", an RDF storage and query processor library for mobile devices [6]. Our framework not only allows storage and querying of RDF data, but more importantly makes it much more user friendly to manipulate the data, be it contextual data from the phone or data from a remote endpoint.

Unlike applications such as Cinemappy [7] and DBPedia Mobile [8], which require developers to have extensive knowledge of both Linked Data and mobile programming, our target audience is Linked Data developers who want to develop mobile applications but do not have mobile programming experience.

5 Summary

Though mobile devices have become the primary computing and communication platform for users, the number of Linked Data apps available for these devices is insignificant because developing them is currently extremely time consuming.

In this demonstration, we show a mobile application platform that enables mobile Linked Data applications to be quickly and easily developed. Along with leading to applications that are able to leverage structured data, these mobile applications will help get around the classic “chicken-or-egg” situation by being both generators and consumers of Linked Data. The current implementation contains a few limitations including scalability, which our team will be working on. We will also be conducting further user studies to evaluate the usability and usefulness of the platform.

References

1. Tillmann, N., Moskal, M., de Halleux, J., Fahndrich, M., Bishop, J., Samuel, A., Xie, T.: The future of teaching programming is on mobile devices. In: Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education, ACM (2012) 156–161
2. Ermilov, T., Khalili, A., Auer, S.: Ubiquitous Semantic Applications: A Systematic Literature Review. International Journal on Semantic Web and Information Systems **10**(1) (2014)
3. Vieweg, S.: Situational Awareness in Mass Emergency: A Behavioral and Linguistic Analysis of Microblogged Communications. PhD thesis, University of Colorado at Boulder (2012)
4. David, J., Euzenat, J., Rosoiu, M., et al.: Linked data from your pocket. In: Proc. 1st ESWC workshop on downscaling the semantic web. (2012) 6–13
5. Chang, K.S.P., Myers, B.A., Cahill, G.M., Simanta, S., Morris, E., Lewis, G.: A plug-in architecture for connecting to new data sources on mobile devices. In: Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on, IEEE (2013) 51–58
6. Le Phuoc, D., Parreira, J.X., Reynolds, V., Hauswirth, M.: RDF On the Go: RDF Storage and Query Processor for Mobile Devices. In: ISWC Posters&Demos. (2010) 12
7. Ostuni, V., Gentile, G., Noia, T., Mirizzi, R., Romito, D., Sciascio, E.: Mobile movie recommendations with linked data. In Cuzzocrea, A., Kittl, C., Simos, D., Weippl, E., Xu, L., eds.: Availability, Reliability, and Security in Information Systems and HCI. Volume 8127 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 400–415
8. Passant, A.: Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations. In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence. (2010)