Exploiting Linked Open Data for Mobile Augmented Reality

Authors

Vinny Reynolds, Michael Hausenblas, Axel Polleres, Manfred Hauswirth, Vinod Hegde.
Email: firstname.lastname@deri.org
Digital Enterprise Research Institute (DERI), National University of Ireland, Galway

Abstract

In this position paper we discuss future directions for mobile augmented reality applications. In particular we focus on "world reference" augmented reality applications that present the user with content within the mobile browser that augments the reality with information about local points of interest such as historical sights, nearby bus stops and cafes. These applications can be generally characterised by the display of points of interest obtained from the execution of spatial queries on a database of these points. As a result, the user has a very domain-specific browsing experience. i.e., is “locked-into” the content of the specific database. Examining existing architectures of augmented reality applications, we observe issues regarding selection and integration of data sources, utilisation of contextual information, and eventually the browsing experience. To address these issues, we propose to exploit Linked Open Data for (mobile) augmented reality applications.

1. The Future of Mobile Augmented Reality?

Imagine the following scenario: *Anthony tells his mobile device: "I'm hungry and only got 20 bucks - find me something to eat, not far away". The mobile device, equipped with an advanced AR browser, shows three matches in an 500m radius, overlaid on the image the device's camera captures. A local Indian restaurant is prioritised above others by the browser which has built up a profile of Anthony's favourite cuisines after previous searches. However today Anthony is feeling adventurous and decides against the restaurant which was highlighted. He asks the AR browser: "anything around my friends would recommend?". The device pulls in restaurant reviews from Anthony's contacts and comes up with a new proposal: a nice Vietnamese restaurant, some 5min away. Anthony walks down the road, holding the device towards the restaurant. He remembers that recently there were some hygienic issues reported regarding some restaurants, downtown. Just to make sure he asks his device: "anything to worry about here?". The AR browser queries Public Sector Information provided by the town and the state and reports back to Anthony. The device shows two restaurants in the same street that had been shut down last week due to health inspection, but not the Vietnamese restaurant Anthony fancied. Now, Anthony is happy and relieved and has a decent meal there.*

Is this scenario possible with the current AR world browsers? In order to answer this question, let us next examine the typical architectures of these devices exploring their features, let us make some observations on their limitations and how finally present our thoughts on how these limitations can be overcome.

2. State of the Art

Figure 1: i) A typical system architecture for a "world browser" AR application. ii) A screenshot displaying available data layers from the popular AR browser Layar. iii) A screenshot from an AR application displaying local properties for sale from a popular Irish website daft.ie.

There are several well known world browser style AR applications. Layar[1] and Wikitude[2] are probably the two most well known, serving mostly reference data and travel data respectively. Junaio[3] and Tagwhat[4] are two examples of more user-centric world browsers allowing clients to tag and upload content as they encounter it in the physical world and to share and discover the content that other users have uploaded.

If we boil down the typical architectures of such systems to their basest components, shown in Figure 1 part i, we can find that these architectures usually comprise of three parts: The user's AR Browser, an AR server that acts as a gateway and finally, a Point of Interest (POI) server that manages and stores the content. In some very domain specific AR applications, the AR server and the POI server may reside at the same location.

An AR browser offers categories or channels of information typically to the user. Upon selecting a channel, the browser will send a query to a server requesting local POI's for that particular channel. The query typically is based around a location and a bounding range. The AR server's primary role is to mediate between the browser and the POI server, similar to a portal. On receiving a query, the AR server usually redirects the query to the appropriate POI server, which then determines the POI's in response to the query and returns the content, via the AR server, to the AR browser .

We observe that in current POI servers for AR applications, they typically serve one specific AR domain, and that there is not a lot of reuse of openly available datasets, e.g. Web data. Also, APIs to query and access remote data are very heterogeneous, e.g. java(script) json APIs are quite very popular, but again, even when using that popular approach, each AR browser and servers typically have their own object models. Recently, moves have been made to try to standardize the way POIs are described by defining an AR specification, ARML[5], however this effort is still at a very early stage.

Moreover, in such typical architectures of existing augmented reality applications, we make the following key observations:

1. selection and integration of data sources is of a static nature and does not scale;
2. contextual information is under-utilised;
3. browsing experience does not support discovery and exploration of new data.

Selection and integration of data sources - At the moment, an AR broker selects and integrates data from different sources in a rather static and non-scalable way. There is little or no interaction between individual reference data sets. For example, if a developer chooses to implement a mash up of two different data sets, for example, the location of nearby restaurants and the location of transportation means near those restaurants a new data set has to be created that combines these two. However, even in this case, the only link between a restaurant and a nearby bus station is their proximity to each other. There is no actual link between the two points, whether that link is a symbolic relationship or even just a hyperlink relating the two points.

Utilisation of contextual information - Only recently, smart phones such as Android phones or iPhones are shipped with a range of sensors such as GPS, motion, etc. The location provided by this sensor has been exploited in pervasive and mobile computing scenarios for over 20 years, and now more recently in mobile augmented reality applications. A typical query to a point of interest server will include the device's current location and a range in which all points will be returned. We note that there is a much wider range of contextual information on a mobile device that might also influence the result of a query to a point of interest server if a query was able to support such expressiveness. This contextual information could include some sensor information that can indicate whether the user is walking or cycling for example, recent social networking on a device can be used to indicate close friends or colleagues, whilst calendar events can signal that the user might be at a certain location at a certain time in the future.

Browsing Experience - Currently, the browsing experience in mobile AR applications doesn't really support exploration and discovery of new content. Consider a typical experience within your current Web browser. The deep hyperlinking of pages naturally supports the discovery and exploration of related material. As you read a Web page, the contained hyperlinks present opportunities to read related pages. In general, this is not the case within an AR browser. The reality is that a AR browser displaying reference points is more akin to using a spatially aware yellow pages than a genuine browsing experience that allows you to follow a hyperlink from one point of interest to another.

3. Linked Open Data to the Rescue ?

Through the application of the Linked Data principles[6] to open datasets (with leading contributors such as BBC, NY Times, Newsweek, US and UK government, etc.) more than 20 billion data items have been made available in the Linked Open Data (LOD) cloud since 2006. The Linked Data principles are:

- **Use URIs as names for things** - each point of interest, each topic, etc. should have its own URI, hence being uniquely named and referenceable by any application, be it an AR browser or a broker;

- **Use HTTP URIs**, so that people can look up those names. In addition, each data item should be accessible from the Web, without requiring additional protocols or tools, rather than a browser (either HTML browser or RDF browser/agent);
- When **someone looks up a URI, provide useful information**, using Web of Data standards (RDF, SPARQL). This would help one to get information about a data item by simply dereferencing its URI (i.e., retrieving it using a software agent). While current practices imply querying a database (often closed) and then translating this information to a generic format, this method will provide a uniform way to get to the desired information;
- **Include links to other URIs** so that they agents can discover more things. This last step will enable interlinking on a global scale.

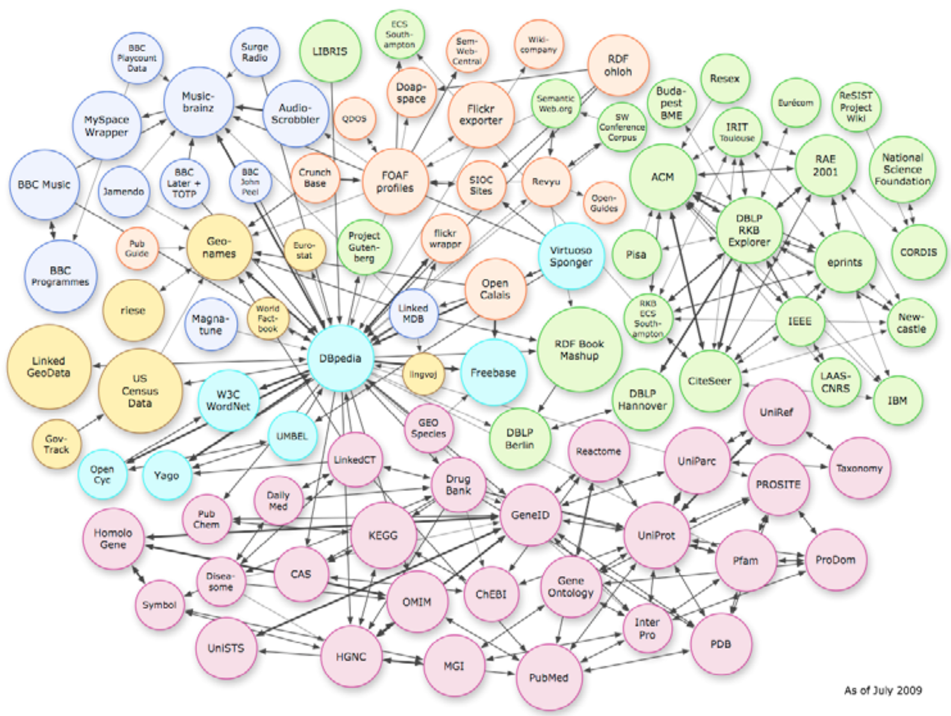

Figure 2: The Linked Open Data cloud, July 2009. Source: Richard Cyganiak and Anja Jentzsch.

The "LOD cloud" already contains plenty of (geo)-location data items, such as found in GeoNames[7], LinkedGeoData[8] or DBpedia[9]. GeoNames, for example, not only provides geographic representation features (such as coordinates), but it includes facts about some places (such as their populations) and it also provides links to DBpedia.

How can the Linked Data in general and the LOD cloud in particular be exploited to address the issues of AR applications?

1. Linked Data supports the **dynamic selection and integration of data** from different data sources that allow for scaling to the size of the Web due to its three pillars: URIs for global unique identifiers of data items, HTTP for an agnostic and reliable access protocol and RDF providing a uniform, graph-based data model.

2. Using the data in the LOD cloud enables to **utilise a wide range of contextual data information** from the AR browser. Beside the obvious location data (e.g. usable in the GeoSPARQL query service[10] , the entry point to LOD sources can virtually be anything. For example, there are dedicated lookup services, such as provided by DBpedia as well as generic Web of Data indexer, like Sindice[11] .
3. The browsing experience with **Linked Data inherently is "Webish"**, see for example the relfinder[12] demo.

4. Discussion Points and Conclusion

We motivated this position paper with a desirable mobile AR application scenario and argued how LOD can be utilised to address current shortcomings. We note, however, in order to realise this, there are some more issues to resolve, we have identified, amongst others, the following issues:

From the end-user perspective - using data from the LOD cloud, especially in a generic fashion, can potentially yield to an information overload. One needs to be able to filter and group entities and topics. This could be addressed by employing reasoning over the data, to map concepts or to group entities. Further, user-based selection can take place; an end-users preferences, the social network and sensor data can be facilitated to harness the data torrent from the LOD cloud.

From the developer perspective - RDF and SPARQL are powerful technologies enabling integration and structured queries of LOD data, respectively. However, both are often perceived being too complex, introducing barriers and slowing down adoption. Hence, supportive activities such as defining minimal subsets of SPARQL executable efficiently for mobile/AR applications, or specific tailored APIS such as the Linked Data API[13] are needed, providing developers access to the LOD cloud in their native environments such as JSON, etc. Also, one needs to find the right tradeoff, whether SPARQL queries should be executed directly on the device running an AR application (in our vision, usually low resource mobile devices), or in how far the necessary data can be preprocessed on the server side.

A cross-cutting concern - relevant to both end-users and developers - exists regarding data provenance and along with it trust. Although Linked Data comes with a sort of built-in provenance mechanism (through URIs and the DNS), the question remains: which data (sources) can and should be used and trusted. Existing efforts such as the W3C Provenance Incubator Group[14] can be used as a starting point, however, much more deployment experience and possibly also research is needed in this area.

We envision that if the W3C decides to launch new activities related to AR applications based on open linked data, liaisons with the ongoing efforts in this regard in the Semantic Web activity (LOD, SPARQL, Provenance Incubator Group) are established and will play a crucial role for defining **open standards and APIs** for AR applications **based on open Web data**.

Acknowledgments

This position paper is a collaboration effort between members of the Linked Data Research Centre (LiDRC[15]) and the Mobile Application Research Centre (MARC[16]), and has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (LION-2)

---

1 http://www.layar.com/
2 http://www.wikitude.org
3 http://www.junaio.com
4 http://www.tagwhat.com/
5 http://www.openarml.org/
6 http://www.w3.org/DesignIssues/LinkedData.html
7 http://www.geonames.org/
8 http://linkedgeodata.org/
9 http://dbpedia.org/
10 http://geosparql.appspot.com/
11 http://sindice.com/
12 http://relfinder.semanticweb.org/
13 http://code.google.com/p/linked-data-api/
14 http://www.w3.org/2005/Incubator/prov/
15 http://linkeddata.deri.ie/
16 http://marc.deri.ie/