Integrating linked open data in mobile augmented reality applications - a case study

Silviu Vert [1], Radu Vasiu [1]

[1] *Politehnica University Timisoara, Piata Victoriei No 2, Timisoara, Romania*

Abstract **– In this paper we develop a model and implement a prototype that integrates user-generated linked open data and open government data in a mobile augmented reality tourism application that works in the browser of the mobile device. We present some challenges met in this endeavour and solutions to overcome them, as well as propose further issues that should be addressed by research in this field.**

Keywords **– Augmented reality, linked open data, open government data, tourism.**

1. Introduction

Augmented reality is a set of technologies, sometimes also called a medium [1], that works as an intuitive interface for people doing some specific activities. It has been proven to be useful in multiple fields, one of them being the tourism domain [2], which is also the focus of our research. Tourism is a significant industry, even in times of crisis [3], that benefits lately more and more from the boom in gadgets and technology. Mobile augmented reality is a suitable approach for delivering a better touristic experience, both of them being based on visual discovery. This benefic combination has been speculated in many mobile augmented reality applications that have been developed with tourism cases in mind [4]. However, these applications are rather static, due to the nature of the content that they process and store. They either use closed databases of information (that is gathered and processed for that application only) or open (but single and disconnected) databases called channels (such as Wikipedia). This is perceived as a limitation by the tourists [5], who are more and more accustomed to dynamic, context-sensitive and adaptive information which they also expect to see in such applications.

Integrating linked open sources of information has been proposed as a suitable solution to overcome such limitations of current mobile augmented reality applications. Linked data [6], the still more practical part of the envisioned Semantic Web [7], works on standards for representing and accessing data that are flexible enough to allow adding and removing sources of data, navigating from one source to another and processing it according to its meaning. As such, linked open data can break the barriers of closed and disconnected information in augmented reality applications. However, this integration is not straightforward and developers have to deal with known linked data integration issues and issues due to its open world assumption, such as geodata integration, data quality assessment, provenance and trust issues [8].

A benefit can be also seen the other way around. There is an intense research in finding suitable ways for visualizing and understanding the increasing amount of linked open data published on the web. One of the targets for this endeavor are lay users, which, as opposed to more experienced ones, need simpler, more intuitive visualization-based browsers for linked open data [9]. It is non-trivial to be able to show intuitive, easy to comprehend and action-ready linked open data and, at the same time, to be able to show large quantities of data from various fields.

There are some early attempts in arguing for the benefits integrating linked open data in mobile augmented reality applications [10] and discussing issues of such integration [11]. Large quantities of data have been integrated in projects from the cultural domain, such as an explorer application for culturally-enhanced Points of Interest (POIs) in Amsterdam [12], and projects in the music domain, such as an application for augmenting posters on the street [13]. A project aiming at overlaying 3D models of buildings on top of the real world [14, 15] demonstrated the mutual benefit of combining augmented reality with linked open data.

While integration issues have been studied in these projects, there is still a lack of research in terms of a model for developing mobile augmented reality applications that integrate multiple sources of linked open data information, with a focus on combining open user-generated information with open governmental data. There is also missing an analysis of how the quality and quantity of available information changes when switching from isolated silos of information to open and interlinked sources of content.

The objectives of this paper are to propose a model and a set of guidelines for implementing a mobile augmented reality touristic application which integrates linked open data and to develop a prototype of such an application. We also aim to analyze the usefulness of integrating multiple sources of linked data information, to highlight obstacles and problems that show up in processing datasets "in the wild" and to propose solutions and ignite discussions based on the findings.

The model we proposed and the prototype we implemented show that there are clear steps in integrating linked open data in mobile augmented reality applications and that this process yields benefits for the tourist in terms of information access.

2. Major components and steps

The model of developing an application that exploits linked data in augmented reality mobile applications resembles closely the architecture of developing a generic linked data application, as described in [16], with some adjustments specific to augmented reality applications. We consider open government data to be a strong pillar in linked open data for augmented reality applications, but, as most of the government data is still only published as raw data, we need to also apply to our model the general Linked Open Government Data lifecycle, as described in [17].

As such, to develop an application that fulfils the objectives set in the introduction, a developer should be aware of the following steps and components.

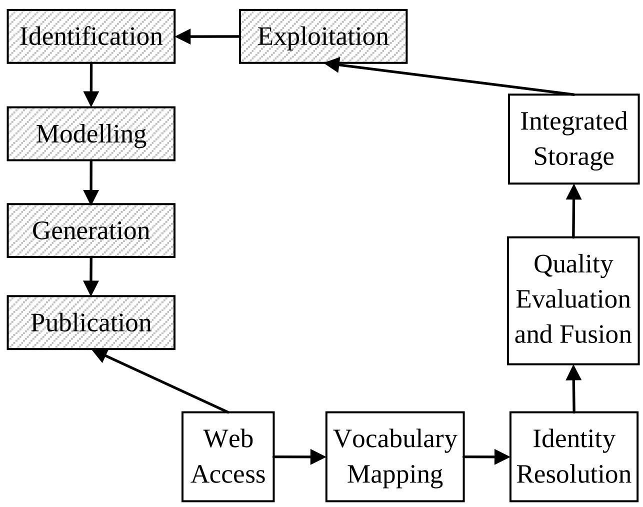

Figure 1. Major steps of the application model

The prototype we developed integrates data from user-generated information sources, like Wikipedia and OpenStreetMap, and from governmental ones, like the Romanian Government Open Data Portal[1]. The former two sources are already published as linked data under the name of DBpedia[2] and, respectively, LinkedGeoData[3]. The governmental source was processed by us to become linked data. We integrate these sources using a linked data integration framework and store the consolidated dataset in a triple store. The geographic area we target is the whole country of Romania, with a focus on the city of Timisoara. We built an augmented reality mobile application that works directly in the browser of the mobile device and which queries the triple store from an Application Programming Interface (API) and shows the information to the user in a typical augmented reality experience.

The steps in the process are detailed as follows.

Identification

The first and foremost important step is identifying the most appropriate datasets to be used for the application's purpose. The selection is based on the quantity of information in the dataset (does it cover the geographic area that I want the augmented reality experience to take place in and does it have a significant number of POIs?), the quality of the information (how complete and accurate is the description of the POIs?) and the attached license (am I allowed to use the information?).

A minimum set of descriptors are required for the POIs to be used in an augmented reality application: label, GPS coordinates and category. While label and GPS coordinates are certainly required to be able to show the POI in space and say what it is, the requirement considering the presence of a category is debatable. We believe that some form of categorizing the POI is strongly needed for proper filtering of the displayed information. Otherwise, there is the danger of showing a cluttered (and, therefore, unusable) display to the user. Other descriptors, not required, but useful to have in an augmented reality environment, are: description (for further information), email/telephone numbers (for the user to be able to email/call the contact person of the POI immediately from his smartphone), opening hours/closing hours (to be able to filter the POIs around her based on the availability at that certain time), picture of the POI (to be able to recognize it easier), links to similar POIs (for the user to be able to further navigate to the other POIs based on similar architecture, functionality etc).

In this paper, we are focusing on open datasets (the ones that are compatible with the Open Definition [18]), from governmental sources and from user-generated ones. To check if a license is truly open, developers can use the list of licenses that are conformant [19] with the specifications written in the Open Definition.

Big linked open datasets can be found easily by using the LOD diagram [20]. The diagram shows what major type each dataset is and the most connected datasets in the world. This visual search is quick and intuitive and works well for identifying

[1] http://data.gov.ro/
[2] http://dbpedia.org/
[3] http://linkedgeodata.org/

big, well-known datasets. Smaller datasets are harder to find this way. One can use tools like DataHub[4] or PublicData[5]. A comprehensive worldwide list of open data catalogs can be found on DataCatalogs[6]. The most recent crawling of the Linked Open Data space [21] shows that geographic datasets occupy 2% of it. Other domains with potential data on POIs are the governmental one, with 18% of the LOD space (the category with, by far, the biggest growth since the last report from 2011) and cross-domain, 4% of the LOD space.

For our prototype, we aimed to identify datasets to be used by tourists visiting Romania. The testing phase would be done in the city of Timisoara, which is currently a candidate for the title of European Capital of Culture in 2021, so it is a place with high touristic potential. Using the above criteria, we identified DBpedia and LinkedGeoData as being suitable candidates. DBpedia covers around 4000 POIs in Romania and LinkedGeoData around 20,000 POIs (the numbers are approximate due to the fact that, for straightforwardness, the crawling of the datasets was done for a circular area centered on the geographical midpoint of the country, so small parts of the neighbouring countries were also included). These datasets have the required descriptors and a few other optional ones as well. Their licenses, Creative Commons Attribution-ShareAlike version 3.0 (CC BY-SA 3.0) and GNU Free Documentation License (GNU FDL) in case of DBpedia and Open Data Commons Open Database License (ODbL) in case of LinkedGeoData, are compatible with the Open Definition.

We could not find appropriate linked open governmental data sources for Romania, so we selected a raw government dataset that was published recently on the Romanian Government Open Data Portal and that lists all the official museums in Romania. The 951 POIs are well described and very appropriate for being used in an augmented reality environment. The dataset[7] was published by the National Institute of Heritage who at the end of 2014 won the Open Government Partnership prize from the Romanian Government for transparency in administration. The portal's license, the Open Government License[8] (OGL ROU 1.0), is compatible with the Open Definition. We believe that the extra work for transforming this raw dataset into a linked dataset is worthwhile. Various valuable raw government data is published currently in many places in the world and the process of transforming it to linked data should be taken into account as a default step by developers.

Table 1. Comparison between various aspects of the integrated sources of data

Name	License	No. of POIs for Romania	Ontology
DBpedia	CC BY-SA 3.0 and GNU FDL	~ 4000	dbo
LinkedGeoData	ODbL	~ 20,000	lgdo
Romanian Open Data Portal	OGL ROU 1.0	951	---

Comparing the identified sources (Table 1 and Table 2) reveals a great variation in licenses, ontologies or form of access, which makes the integration a non-trivial task.

Modelling

The modelling stage concerns directly the governmental data source. This step consists of analysing the information and deciding on ontology to be used in modelling that information. If the other sources of information to be mixed are already identified, it is helpful to use for the governmental source the same ontology as the one which is the most relevant in the other sources. However, in many cases, each source of information has its own ontology, so a suitable approach is to use the ontology of the most important of them.

In our use case, the user-generated data sources use quite different approaches in modelling. LinkedGeoData uses a lightweight OWL ontology, called the LinkedGeoData Ontology (lgdo), which maps pretty directly on the key-value pairs, known as tags, that are specific to POIs in OpenStreetMap [22]. DBpedia uses a shallow, multi-domain ontology, called the DBpedia Ontology (dbo), which maps Wikipedia infoboxes through hand-made rules [23].

For the government data source, we considered a combination of vocabularies (such as FOAF or Basic Geo) but also a small part of the DBpedia ontology, for easier alignment.

Generation

The generation step involves transforming the raw data into linked data and it applies to the government data source. For this purpose, we chose the OpenRefine tool, as it has some very useful features for dealing with data: exploring, cleaning,

[4] http://datahub.io/
[5] http://publicdata.eu/
[6] http://datacatalogs.org/
[7] http://data.gov.ro/dataset/ghidul-muzeelor-din-romania
[8] http://data.gov.ro/base/images/logoinst/OGL-ROU-1.0.pdf

transforming, reconciling and matching [24]. Because the effort was for generating linked data for just one dataset, we chose a manual approach. In case of a significant number of datasets, a (semi-)automatic approach is more suitable, such as the one employed by the csv2rdf4lod tool used for the TWC LOGD portal [25].

Publication

This step refers to the publication of the generated linked dataset. This is not required if the linked data is only to be used in the application. However, the data might be useful to other developers as well, so it is a nice-to-have feature. We will publish the linked government dataset after switching from the development triple store to the production triple store.

Integration (Access + Integration + Storage)

For the integration of the datasets we have chosen the popular Linked Data Integration Framework (LDIF) [26]. This framework covers four modules from the generic schema of linked data applications: the Web Data Access Module, the Vocabulary Mapping Module, the Identity Resolution Module and the Quality Evaluation and Fusion Module. The tool is very useful in such cases as it streamlines the operations of accessing the data, aligning it, merging it and checking its quality. LDIF has configuration files where one can modify parameters to change its runtime behaviour. Other similar tool that developers might use is Apache Marmotta[9].

Integration – Access. The data sources can be accessed in various ways (Table 2).

Table 2. Comparison between accessing methods available for the integrated sources

Name	RDF	SPARQL	REST API	Dump
DBpedia	x	x		x
LinkedGeoData	x	x	x	x
Romanian Open Data Portal			x	x

Linked data applications present three types of data access patterns [16]: the crawling pattern, in which several sources are crawled, the data is fetched and stored in a single datastore, from where it is accessed later; the on-the-fly dereferencing pattern, according to which the data sources are crawled by dereferencing the URIs on the spot and as such going from one to the other in real time; and the query federation pattern, in which a query is sent to a fixed set of data sources that are known beforehand.

The latter two patterns provide the most up to date data; however, they might not provide the data fast enough. This can be an issue in augmented reality environments, where the availability of the data for the registration to happen in real time is a critical aspect of a successful experience. The first pattern might have the disadvantage of not providing the latest data (though a frequent refresh of the data can make this issue a minor one) but the resulting data is available immediately.

In our use case, we favoured the crawling pattern, which is also the one used by the LDIF platform. We specified in the LDIF configuration file how each data source should be fetched, depending on the available data access methods presented in Table 2. DBpedia data was downloaded both through SPARQL queries requesting categories specific to Romanian POIs and through data dumps, while LinkedGeoData and the governmental source were inputted only as data dumps. For the ease of processing, we crawled those POIs from DBpedia and LinkedGeoData that were in a certain radius around the center of the country, meaning that we also got some POIs outside the country (as it does not have a circular geographic area).

Integration – Vocabulary Mapping. LDIF uses internally the R2R framework [27] to achieve the translation of the ontology in the original source to the desired resulting ontology. Looking into the data of our selected sources, we see various categories that the POIs belong to. The dbo ontology uses straightforward categories, such as *SportFacility* or *EducationalInstitution*. However, DBpedia also uses the yago ontology, which has categories that are hard to translate automatically (such as *PlacesOfWorshipInTimisoara* or *SynagoguesInRomania*) due to the way they are built (extending Wikipedia categories were extended with WordNet [28]), which are hard to manually process. Because of this issue, the categories that should be translated must be picked up manually from the data with the desired purpose in mind. LinkedGeoData has some more straightforward categories, such as *FuelStation*, *Parking*, *Supermarket* etc.

Filtering by category is important for the user, but displaying all these niche categories on a small screen might puzzle the user. Moreover, displaying such niche categories would mean that the user might not find anything close to her in several categories and would be disappointed by the augmented reality experience.

[9] http://marmotta.apache.org/

As such, we empirically devised 7 supercategories that would encompass all the smaller categories: Tourist (churches, museums), Utility (fuel station, exchange offices), Food&Drinks (restaurants, pubs), Shopping (supermarkets, shopping malls), Emergency (pharmacies, hospitals), Entertainment (sports, cinema) and Institution (townhall, university).

The translation rules were specified by indicating that the source's category is *rdfs:subClassOf* our own supercategory, which belongs to our ontology named tomldifo. Below are snippets of code showing examples of translation rules for DBpedia and LinkedGeoData categories.

```
yago:PlacesOfWorshipInTimişoara rdfs:subClassOf
tomldifo:tourist .
lgdo:Monument rdfs:subClassOf tomldifo:tourist .
…
dbo:ShoppingMall rdfs:subClassOf tomldifo:shopping .
lgdo:Shop rdfs:subClassOf tomldifo:shopping .
…
dbo:University rdfs:subClassOf tomldifo:institution .
lgdo:University rdfs:subClassOf tomldifo:institution .
```

The museums in the governmental data source were translated to the tourist supercategory.

Generally, the translation step should also involve aligning all the sources to the same geographic vocabulary. It was not the case for our prototype, as all the sources are using the WGS84 system. However, we had to create the *geo:geometry* attribute, from the latitude and longitude attributes, for the governmental source, because this step was not done in OpenRefine, and for data that was missing this attribute in the other sources. This attribute is needed internally by Silk in the next module, Identity Resolution. Below is a snippet of code showing the creation of the *geo:geometry* attribute.

```
mp:toGeometry
a r2r:Mapping ;
r2r:prefixDefinitions "geo:
<http://www.w3.org/2003/01/geo/wgs84_pos#> . type:
<http://www.openlinksw.com/schemas/virtrdf#> ." ;
r2r:sourcePattern "?SUBJ geo:lat ?lat . ?SUBJ geo:long ?long"
;
r2r:targetPattern "?SUBJ geo:geometry
?'geometry'^^type:Geometry" ;
 r2r:transformation "?geometry = concat('POINT(',?long, ' ',
?lat,')')" .
```

Integration – Identity resolution. For the identity resolution, LDIF uses the Silk tool [29] to be able to match and merge identical POIs that appear in multiple sources. Uncertainty in fusing information might yield three main integration issues: geographic integration, place name integration and semantic integration [30]. Correspondingly, we used as criteria for merging: the geographic distance between the POIs, the similarity of the names of POIs (using the Levenshtein distance algorithm) and the supercategory that they belong to.

These were by no means the only criteria one could use for integration purposes. The GPS coordinates of the boundaries of a POI are a piece of very useful criteria, for example, but this kind of information is virtually impossible to find in real user-generated data.

A clear condition we set for two POIs to be considered for integration is that they belong to the same supercategory. For the geographic distance between the POIs and the Levenshtein distance on names, we used empirically determined thresholds. To determine these thresholds, we imported the information of the POIs in Google Fusion Tables [31] and displayed them on the map of Romania, then zoomed in on the area of Timisoara, that we know well. We colour coded the pinpoints of the POIs based on the source of the data, so we were able to spot quite rapidly if certain POIs are susceptible of being integrated based on the geographic distance (Figure 2). We then looked over their other information (category and name) to see how much they differ, in the real world of user-generated data, even if they are the same POIs actually. We favoured the more restrictive thresholds, as we believe it is better for the user to see two POIs that are basically the same then miss one totally because it was merged wrongly with other.

Figure 2. Screenshot of the Google Fusion map with overlaying POIs

Below is a short snippet of code showing the rules for merging POIs belonging to the "institution" supercategory. The threshold for Levenshtein distance is set to 1 (meaning the name can differ by maximum 1 character) and for the geographic distance is set to 100 (meaning it can differ by maximum 100m). The merging is done by creating sameAs links, as in the code snippet below.

```
<Interlink id="institution">
 <LinkType>owl:sameAs</LinkType>

 <SourceDataset dataSource="SOURCE" var="a">
  <RestrictTo>?a rdf:type tomldifo:institution .</RestrictTo>
```

```
  </SourceDataset>

  <TargetDataset dataSource="TARGET" var="b">
    <RestrictTo>?b rdf:type tomldifo:institution .</RestrictTo>
  </TargetDataset>

  <LinkageRule>
    <Aggregate type="average">
      <Compare metric="levenshteinDistance" threshold="1"
required="true">
        <TransformInput function="lowerCase">
          <Input path="?a/rdfs:label" />
        </TransformInput>
        <TransformInput function="lowerCase">
          <Input path="?b/rdfs:label" />
        </TransformInput>
      </Compare>
      <Compare metric="wgs84" threshold="100" >
        <Input path="?a/geo:geometry" />
        <Input path="?b/geo:geometry" />
        <Param name="unit" value="m" />
      </Compare>
    </Aggregate>
  </LinkageRule>

  <Filter />
</Interlink>
```

Integration – Quality Evaluation and Fusion. For evaluating the quality of the integrated data and resolving potential conflicts in data, LDIF uses the Sieve module [32]. This module allows, in the first phase, to assign quality indicators, based on various scoring functions, to the data. The scoring functions can be related to how recent the data was crawled or updated, the reputation of the data (where it comes from) or they can compute indicators based on some numeric properties of the data.

For assessing the quality of the data in our prototype, we used criteria such as time closeness and reputation. We favoured the information that was updated more recently and we favoured the English Wikipedia over the other localized versions of it.

In the second phase, the Sieve module performs the fusion of the data, based on rules set by the developer and which take into account the quality indicators. For our prototype, we chose to keep the GPS coordinates of the most recently updated POI from several POIs that were identified as being the same.

Integration – Storage. For storage purposes, we use an OpenRDF Sesame[10] triple store version 2.7.9 on a Linux 2.6.32 server. There are a total of 1237760 statements currently stored on the server, belonging to POIs in the areas of Trento, Italy (crawled for demo purposes [33]) and the whole Romania.

[10] http://rdf4j.org/

Exploitation

The exploitation step involves building some sort of application or mashup that exploits the integrated linked data. Our implemented prototype is a mobile augmented reality application that works in the browser of the smartphone or tablet. Therefore, the tourist is not required to download a separate application for her purpose. This way, we come closer to the concept of ubiquitous augmented reality.

Because the capability of mobile phones to interact directly with SPARQL endpoints is still in its infancy, even more in case of using just code in the browser, we have built an API on top of the triple store that allows the web application to get the information in JSON format. The web application can make two kinds of requests on the API:

a) get the list of POIs (with name and GPS coordinates) based on the position of the tourist, the desired radius (which is actually an almost-square rectangular area, due to poor geospatial support in the triple store [34]) and the desired type of POIs (supercategory)

b) complete information about a certain POI, meaning name, description, picture, original source and link to further information

The application works like this (see also Figure 3):

1) the tourist opens the web application in his mobile browser and can see pinpoints overlaid on real POIs around him. The pinpoints have different icons, depending on the category the POIs belong to, so it is easy for the tourist to understand at a glance what is around her. There is a menu available all the time on the top right corner of the screen, from where the user can choose to see all or just one of the supercategories and can adjust the radius for displaying the POIs.

2) the tourist taps on a pinpoint and a small window opens on the bottom of the screen, showing the name of the POI, the distance to it and a link for complete information.

3) if the tourist presses that link, a bigger window opens, covering almost all the screen, and displaying complete information on the POI (name, description, picture, source, link to webpage).

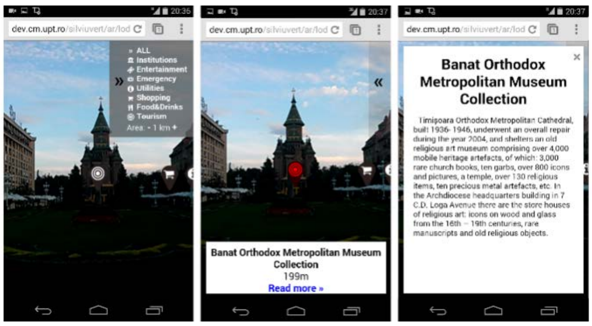

Figure 3. Screenshots of the web application

The web application is built on top of awe.js, an augmented reality web library[11] that uses technologies like HTML5, CSS and Javascript, combined with access to modern capabilities of mobile devices, such as geolocation, photo/video camera, WebRTC or WebGL, to show 2D or 3D augmented reality objects on top of the surroundings. The awe.js library works on top of three.js, a library that allows developers to design 3D experiences in the browser. According to the developers of awe.js, this library should work on the latest versions of Chrome and Firefox on Android, as well as on devices such as Leap Motion, Oculus Rift and Google Glasses. We successfully tested the application on various Android smartphones and tablets.

3. Discussion and further work

The prototype that we have implemented, highlighted some well-known linked data integration issues, but also issues specific to our use case.

Sources differ greatly in terms of vocabulary used, form of access, license and so on, so proper identification and analysis is needed to prepare them for further processing. Tools and guidelines for this step are available and some have been presented here.

A significant manual effort is needed for translating the data. For example, some categories are titled in such a way (e.g. *TypeOfBuilding*In*City*) that it is impossible to know them beforehand. As such, developers have to look thoroughly into the data and pick up those information manually as best as they can.

Popular user-generated sources of information have a large quantity of information available for almost every geographic area. They are also great in linking datasets between them. However, the POIs that are not in the top of the interestingness ladder, that is most of them, are usually described shallowly. Governmental sources can be much more detailed – usually, if one element in the list is well-described, so are all the others. Their downside is that, with rare exceptions, the data is neither translated nor linked.

Geographic support is still poor in current triple stores, despite that the GeoSPARQL standard has been launched in 2012. We could not request the POIs that are in a circular geographic area centered on the position of the tourist, due to the fact that the OpenRDF Sesame triple store does not implement this feature. We instead requested the POIs that are inside an almost-square rectangular area centered on the position of the tourist. This approach barely hinders the experience of the tourist, however, it is certainly more expensive in terms of the request's response time. It is advisable that the production server use a triple store that implements SPARQL requests on a circular geographic area, such as Virtuoso[12].

The data is accessed in real-time, which is fine for those that have a 3G connection. This happens in the case of most citizens and tourists from the same country. Foreign tourists are less probable to have a 3G connection, so a solution must be searched in this case (e.g. downloading a consolidated dataset for one city at a time, before visiting it).

Names and geographic position vary greatly between POIs in different information sources that are actually the same, this being a classical issue of user-generated data. We observed that it is almost impossible to set thresholds for detection algorithms that keep both the number of missed unmerged similar POIs and the number of false positives low. For example, there are identical POIs that differ by hundreds of meters in geographic distance and by several words in the title (e.g. "Timisoara Zoological Garden" in DBpedia vs "Padurea Verde Zoological Garden" in the governmental dataset, separated by 213m, or "Stadion CFR" in LinkedGeoData vs "Stadionul CFR (Timisoara)" in DBpedia, reported as being 101m away one from the other). Setting such large thresholds, that would enable proper merging of identical POIs, would undoubtedly trigger false positives. Of course, we also encountered situations where identity resolution is easy to do, such as for the POIs "Museum of Banat" in DBpedia and "Museum of Banat" in the governmental dataset, with a reported distance between them of 25m. The algorithms are hampered by the fact that DBpedia usually has labels in English and LinkedGeoData in Romanian, for the geographical area of Romania.

These settings for integration can be improved by including additional datasets in the process, extending the targeted geographic area and analysing the output. The application can benefit from an automatic discovery and inclusion of governmental datasets, which is a field where research is taking place nowadays. Further analysis is useful in the quality evaluation step, as the information that guides a tourist in real-time in an unknown place is regarded as being crucial for a successful experience.

Regarding the augmented reality web application, we demonstrated the possibility of showing this integrated data as augmented reality directly in the browser, thanks to newer capabilities of mobile devices. However, the user interface can be improved by, for example, showing subcategories inside

[11] https://github.com/buildar/awe.js

[12] http://virtuoso.openlinksw.com/

supercategories, for better filtering, or showing links to similar POIs (regarding architecture, functionality and so on) for a more complete experience.

4. Conclusion

Tourists nowadays expect to be able to use dynamic, adaptive and personalized technology applications to move around an unknown place. Mobile augmented reality applications are an appropriate way for tourists to explore new lands. However, these applications suffer in terms of content, as information is rather limited and not connected in the sense of multiple sources available. Linked open data seems to be a good solution for this problem, on the one side, due to its openness and increasing value in quantity and quality, and on the other side, due to its dynamic and easy way of interconnection.

We proposed a model for building such an application and developed a prototype that implements it. We showed how each step of the process, regarding both linked data and augmented reality, can be handled, in a sufficient generic manner, which can be replicated by other researchers. We pointed out that integrating linked data in mobile augmented reality applications has the potential of leading to more and better data for the tourist. We have discussed what type of problems can show up, how to overcome them and what are some next steps in improving this kind of applications.

Acknowledgements

This work was partially supported by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of National Education, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

References

[1]. Craig A.B. (2013). *Understanding Augmented Reality: Concepts and Applications*. Newnes

[2]. Huang Z., Hui P., Peylo C., & Chatzopoulos D. (2013). Mobile Augmented Reality Survey: A Bottom-up Approach. arXiv preprint arXiv:1309.4413

[3]. Franco C.L. (2010). CONCERT: A New Framework for Contextual Computing in Tourism to Support Human Mobility. UNIVERSIDAD DE DEUSTO (SPAIN)

[4]. Yovcheva Z., Buhalis D., & Gatzidis C. (2012). Overview of Smartphone Augmented Reality Applications for Tourism. E-review of Tourism Research 10:

[5]. Langlotz T., Nguyen T., Schmalstieg D., & Grasset R. (2014). Next-Generation Augmented Reality Browsers: Rich, Seamless, and Adaptive. *Proceedings of the IEEE* 102:155–169.

[6]. Bizer C., Heath T., & Berners-Lee T. (2009). Linked Data - The Story So Far: *International Journal on Semantic Web and Information Systems* 5:1–22. doi: 10.4018/jswis.2009081901

[7]. Shadbolt N., Hall W., & Berners-Lee T. (2006). The semantic web revisited. *Intelligent Systems, IEEE* 21:96–101.

[8]. Vert S., & Vasiu R. (2014). Relevant Aspects for the Integration of Linked Data in Mobile Augmented Reality Applications for Tourism. In: Dregvaite G, Damasevicius R (eds) Information and Software Technologies. Springer International Publishing, pp 334–345

[9]. Dadzie A.-S., & Rowe M. (2011). Approaches to visualising linked data: A survey. *Semantic Web* 2:89–124.

[10]. Reynolds V., Hausenblas M., Polleres A., Hauswirth M., & Hegde V. (2010). Exploiting linked open data for mobile augmented reality. W3C Workshop: Augmented Reality on the Web. June, p 12

[11]. Hegde V., Reynolds V., Parreira J.X., & Hauswirth M. (2011). Utililising Linked Data for Personalized Recommendation of POI's.

[12]. Van Aart C., Wielinga B., & Van Hage W.R. (2010). Mobile cultural heritage guide: location-aware semantic search. Knowledge Engineering and Management by the Masses. Springer, pp 257–271

[13]. Nixon L.J.B., Grubert J., Reitmayr G., & Scicluna J. (2012). SmartReality: Integrating the Web into Augmented Reality. pp 48–54

[14]. Aydin B., Gensel J., Genoud P., Calabretto S., & Tellez B. (2013). Extending Augmented Reality Mobile Application with Structured Knowledge from the LOD Cloud. 3rd International Workshop on Information Management for Mobile Applications 13:

[15]. Aydin B., Gensel J., Genoud P., Calabretto S., & Tellez B. (2013). An architecture for surroundings discovery by linking 3D models and LOD cloud. Proceedings of the Second ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems. ACM, pp 9–16

[16]. Heath T., & Bizer C. (2011). Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology* 1:1–136. doi: 10.2200/S00334ED1V01Y201102WBE001

[17]. Villazón-Terrazas B., Vilches-Blázquez L.M., Corcho O., & Gómez-Pérez A. (2011). Methodological guidelines for publishing government linked data. Linking Government Data. Springer, pp 27–49

[18]. Open Knowledge The Open Definition. http://opendefinition.org/. Accessed 6 Sep 2014

[19]. Open Knowledge Conformant Licenses - Open Definition. http://opendefinition.org/licenses/. Accessed 14 Jan 2015

[20]. Cyganiak R., & Jentzsch A. (2011). Linking open data cloud diagram. LOD Community (http://lod-cloud. net/)

[21]. Schmachtenberg M., Bizer C., & Paulheim H. (2014). Adoption of the linked data best practices in different topical domains. The Semantic Web–ISWC 2014. Springer, pp 245–260

[22]. Stadler C., Lehmann J., Höffner K., & Auer S. (2012). Linkedgeodata: A core for a web of spatial open data. *Semantic Web* 3:333–354.

[23]. Bizer C., Lehmann J., Kobilarov G., Auer S., Becker C., Cyganiak R., & Hellmann S. (2009). DBpedia-A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7:154–165.

[24]. Huynh D., & Mazzocchi S. (2012). *OpenRefine*.

[25]. Ding L., Lebo T., Erickson J.S., DiFranzo D., Williams G.T., Li X., Michaelis J., Graves A., Zheng J.G., Shangguan Z., Flores J., McGuinness D.L., & Hendler J.A. (2011). {TWC} LOGD: A portal for linked open government data ecosystems. *Web Semantics: Science, Services and Agents on the World Wide Web* 9:325 – 333. doi: http://dx.doi.org/10.1016/j.websem.2011.06.002

[26]. Schultz A., Matteini A., Isele R., Mendes P.N., Bizer C., & Becker C. (2012). LDIF-A Framework for Large-Scale Linked Data Integration. 21st International World Wide Web Conference (WWW 2012), Developers Track, Lyon, France

[27]. Bizer C., & Schultz A. (2010). The R2R Framework: Publishing and Discovering Mappings on the Web. Proceedings of the First International Workshop on Consuming Linked Data, Shanghai, China, November 8, 2010

[28]. Suchanek F.M., Kasneci G., & Weikum G. (2007). Yago: a core of semantic knowledge. Proceedings of the 16th international conference on World Wide Web. ACM, pp 697–706

[29]. Volz J., Bizer C., Gaedke M., & Kobilarov G. (2009). Silk-A Link Discovery Framework for the Web of Data. LDOW 538:

[30]. Karam R., Favetta F., Kilany R., & Laurini R. (2010). Integration of similar location based services proposed by several providers. Networked Digital Technologies. Springer, pp 136–144

[31]. Gonzalez H., Halevy A., Jensen C.S., Langen A., Madhavan J., Shapley R., & Shen W. (2010). Google fusion tables: data management, integration and collaboration in the cloud. Proceedings of the 1st ACM symposium on Cloud computing. ACM, pp 175–180

[32]. Mendes P.N., Mühleisen H., & Bizer C. (2012). Sieve: Linked Data Quality Assessment and Fusion. Proceedings of the 2012 Joint EDBT/ICDT Workshops. ACM, New York, NY, USA, pp 116–123

[33]. Vert S., Dragulescu B., & Vasiu R. (2014). LOD4AR: Exploring Linked Open Data with a Mobile Augmented Reality Web Application. Proceedings of the ISWC 2014 Posters & Demonstrations Track, within the 13th International Semantic Web Conference (ISWC 2014). Riva del Garda, Italy, pp 185–188

[34]. GeoKnow Consortium (2013). Deliverable 2.1.1 - Market and research overview.

Corresponding author: Silviu Vert
Institution: Politehnica University Timisoara,
Piata Victoriei No 2,
Timisoara, Romania
E-mail: silviu.vert@cm.upt.ro