Linked Data in Action: Personalized Museum Tours on Mobile Devices

Olga Kovalenko[1], Yassine Mrabet[2], Kim Schouten[3], and Suad Sejdovic[4]*

[1] Vienna University of Technology, Austria
kovalenko@ifs.tuwien.ac.at
[2] Luxembourg Institute of Science and Technology, Luxembourg
yassine.mrabet@list.lu
[3] Erasmus University Rotterdam, the Netherlands
schouten@ese.eur.nl
[4] FZI Research Center for Information Technology, Germany
sejdovic@fzi.de

Abstract. Producing and maintaining museums audio guides is both time consuming and expensive. Typically the existing methods to build tour guides lack either the personalization aspect or the scalability aspect related to the ad hoc data format used to represent the exhibited objects. In this paper we present *Living Museum*, a free mobile application that assists museum visitors with recommendations and multimedia description of the visited artifacts. Our approach takes into account the location of the visitor and his profile that is incrementally built based on visited artifacts. We argue that Linked Open Data allows building repeatable methods and present a first prototype using the British Museum's SPARQL endpoint. We provide general feedback on implementing semantic-web enabled applications on mobile devices and specific feedback on manipulating the British Museum dataset.

1 Introduction

Over the years, museums, such as Le Louvre or the British Museum, have acquired huge collections of artworks from different cultures and historical periods that provide an amazing experience to the interested visitor. The huge number and variety of art objects can be stunning as well as overwhelming, or even a little frightening, for the unprepared visitor. The pulsating question when entering a museum is: "Where shall I go to and how do I navigate this maze of rooms, stairs and corridors?".

Although technology drastically evolved over the last decades, the level of support typically provided to museum visitors did not see a similar improvement. Typically one can choose between three options: (1) an *audio guide*, that provides a recorded spoken commentary and is normally realized on a hand-held device. The provided information is limited and the descriptions quality varies strongly

* All authors contributed equally to this work.

per specific museum/exhibition. This is a cheap solution, but it lacks flexibility and convenience; (2) joining a *visitors group* with a human guide. This is more expensive, the quality of explanations depends very much on a particular guide, the tour logistics is typically predefined, but as an advantage one can ask extra questions during the tour; (3) taking a *personal guide*: a person competent in a particular exhibition, historical period or subject, will lead a visitor through the museum. This yields both flexibility and high quality explanations, but at a high cost. As such, this is only available to a select number of visitors.

Nowadays, a plenty of data is published on the Web and is available for use, including information on objects exposed in museums. The apparent next step is integrating this data into guided tour solutions for museums. Potentially useful data sources range from general data sources like DBPedia to specific data sources in the cultural heritage domain.

In this paper we present an application for mobile devices that provides a personalized guided tour through the various exhibitions of the British Museum. The presented solution takes features of the displayed art works into account (e.g., historical period, creator, culture, etc). Furthermore, it uses the floorplan of the museum to filter objects that are not far away from current visitor location. The application responses interactively to the user feedback: new recommendations are generated on the fly based on a visitor's history of "likes" and "dislikes" from already seen objects. Also, an audio description for an art object can be automatically produced from the corresponding textual description. The application can be installed on the visitor's smartphone, a device that most visitors already have, and that is able to play audio and display video or images.

For the demonstration within this paper we used the British Museum dataset[1] that contains rich information about the objects presented at the exhibitions of British Museum. Available data for the art object covers its title; current location inside the museum (up to a specific room); to which ethnic group, geographical place, person or event it relates; its composing material, etc. The data is represented in the form of RDF triples and is published as Open Linked Data. A SPARQL endpoint[2] is provided to query the dataset from external applications. The dataset is organized according to CIDOC Conceptual Reference Model [4] to make it interoperable with other cultural heritage data. Besides CIDOC, it also uses SKOS [5] constructs to relate different terminologies within the dataset. However, the presented solution can work in general for any museum, if a dataset with enough detailed descriptions for art objects is provided.

2 Application Walkthrough

An overview of the architecture of the application can be found in the figure below. Note that the use of external linked open data is not yet implemented in our prototype. Corresponding source code is available on the Github [3].

[1] `http://datahub.io/dataset/british-museum-collection`

[2] `http://collection.britishmuseum.org/sparql`

[3] https://github.com/suads/BM_LivingMuseum

The app starts by giving the three possible starting artifacts for a personalized tour. As British Museum has multiple entrances, this first recommendation takes into account the current location of the user to provide starting objects that are not far away. The choice of a starting object is the first cue for the app to gauge the user's interest. During the tour, the user has access to two tabs: the *description tab* containing a description of the current object and the *recommendation tab* showing possible next objects to view.

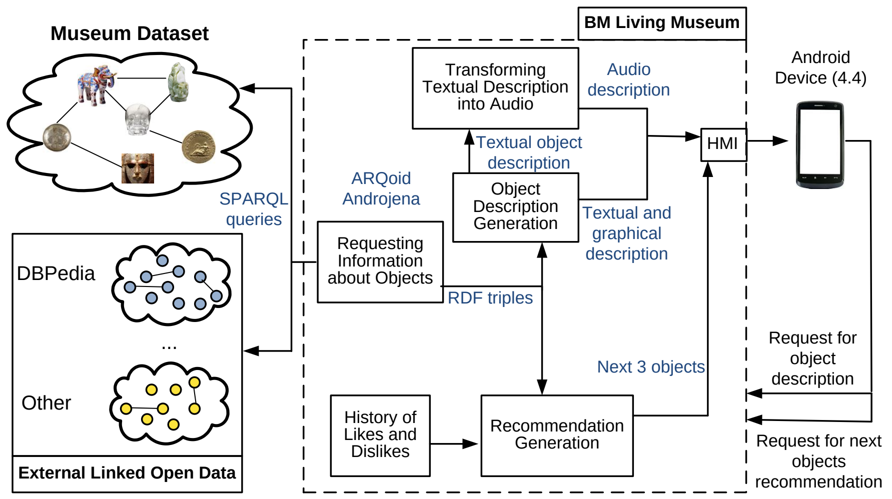
Museum Dataset
BM Living Museum
Transforming Textual Description into Audio
Audio description
Android Device (4.4)
HMI
Textual object description
ARQoid Androjena
Object Description Generation
SPARQL queries
Textual and graphical description
Requesting Information about Objects
DBPedia
RDF triples
...
Next 3 objects
Request for object description
Other
History of Likes and Dislikes
Recommendation Generation
Request for next objects recommendation
External Linked Open Data

Fig. 1: Overview of the app's architecture.

On the *description tab* one can view information drawn from the museum's own data set, including a textual description and a picture. If available, information from external data sources (e.g., DBPedia), can be integrated into the description. Furthermore, an audio description is available, by means of Android's built-in library for speech generation. The user can like or dislike the current object, helping the app to generate useful suggestions.

On the *recommendation tab* the three next potential objects to visit (with picture and location) are presented to navigate user through the exhibition.

3 Recommendation Algorithm

Artifacts in the current visitor location and neighboring rooms are potential recommendation targets: the New Location Set (**NLS**). This dynamic artifacts set is ordered according to the similarity of each artifact with the visitor profile.

The **visitor profile** consists of a multi-dimensional vector representing the features of the previously visited artifacts. The value of each feature/dimension is an integer that is incremented and decremented according to the number of visits and like/dislike actions. The **similarity** of an artifact and a visitor profile is the cosine similarity of the vector representing the visitor interests and the artifact features (with values set to 1 for the artifacts vector).

At each artifact visit and/or like/dislike action the user profile is updated and the artifacts list is re-ordered. We consider only the N last visited artifacts to build the user profile. At N+1, the oldest artifact is removed from the visitor history as well as its impact on the profile vector. This allows adapting the recommendations to changes in users interests over time. For instance, if the visitor liked more than 20 artifacts related to Ancient Egypt in the first part of his visit then started looking for artifacts related to the Byzantine Empire, considering only the 5 last visited artifacts will allow a quick shift in the type of recommendations. In our experiments N was set to 10. The artifact list is also updated when the visitor enters a new room in the museum (c.f. NLS) and re-ordered according to the similarity of the (new) artifacts with the user profile. When the user press the recommendations button he gets the 3 most similar artifacts at that moment.

4 Semantic Web Technologies on Mobile Devices

One of the driving forces behind the proposed prototype is to explore the technical limitations of mobile devices when using Semantic Web Technologies as well as to gain experience regarding the usability for app developers without a professional infrastructure (e.g. a data center) to delegate all the calculation and querying to the backend. Another aspect to explore was which challenges may arise when using Open Linked Data, considering not only technical challenges, but also challenges related to content (e.g. completeness of data).

From the technical perspective we used a OnePlus smartphone with 3 GB of RAM and a Qualcomm Quad-Core CPU with 2.5 GHz running Android 4.4.4. The app was implemented for the Android operating system with a minimum level of 19 (4.4). For accessing the semantic information we made use of the Androjena Project [1], a port of Apache Jena to the Android platform. Unfortunately, the latest release of Androjena was in 2010 and development has stopped since then. Thus, the Androjena project supports only SPARQL 1.0, which fortunately was enough for our use case. The Androjena port consists of one main library "androjena_0.5", which contains original Jena 2.6.2 source code modified to run on Android. Additionally, the following libraries are shipped with Androjena: 1) IBM ICU4J 3.4.5: International Components for Unicode; 2) Jena IRI 0.8: Support for Internationalized Resource Identifiers in Jena; 3) Apache Xerces: Collection of libraries for working with XML; 4) SLF4J Android: Port of SLF4J logging framework to the Android platform.

In addition to Androjena we used ARQoid[2], the ported version of Jena query engine for SPARQL. For query execution over the endpoint we used "QueryExecution" (for single query execution) and "QueryExecutionFactory" (to create "QueryExecution" objects from a query). The "QuerySolution" interface was used together with "ResultSet" to get an answers set for a query.

As the projects that aimed to port ARQ and Jena to Android have ceased activity in 2010 leaving the software unfinished, we had to make a few modifications to the documented approach for using them. On top of Androjena and ARQoid we imported Apache Jena 2.11, manually removing some of the jars from the build path as they were not compatible to the Dalvik virtual machine, which is used as the Java runtime environment in Android.

5 Lessons Learned and Discussion

The British Museum dataset was not entirely complete and regular w.r.t. the schema; i.e. for some artifacts not all specified properties are available. For instance, some artifacts have no picture or the short synopsis is missing. Also, the artifact title is not always accessible with the same query pattern. We identified two main title access methods (in terms of SPARQL triple patterns):

```
{ ?ep crm:P70_documents ?obj . ?ep dc:title ?title . }
UNION { ?obj crm:P128_caries ?x . ?x rdfs:label ?title . }
```

To deal with the dataset irregularity and incompleteness we split queries into several optional clauses, except for the location of the artifact, which is mandatory for our approach. Also, as several locations are not rooms dedicated to artifacts (e.g. stairs, halls) we looked particularly for location values containing the string "ROOM_ID". We also observed that some locations are present in the dataset, but are not represented in the actual museum's floorplan.

```
...
SELECT DISTINCT ?obj ?ep ?title ?image ?desc ... {
?obj crm:P55_has_current_location ?loc .
?loc rdfs:label ?location .
FILTER ( regex(?location, ".*ROOM_ID.*", "i") )
OPTIONAL {{ ?obj crm:P102_has_title ?t0 . ?t0 rdfs:label ?title .}
UNION { ?obj crm:P128_caries ?x . ?x rdfs:label ?title . } }
OPTIONAL { ?obj bmo:PX_has_main_representation ?image . }
OPTIONAL { ?obj bmo:PX_physical_description ?desc . }
...
```

From a querying point-of-view, using SPARQL from the mobile application is the most efficient option. Using other protocols, e.g. Web services using JSON [3], leads to additional networking cost with a backend server, which will have to first query the SPARQL endpoints and then send the results to the application. However, current APIs to directly query SPARQL endpoints from mobile devices are neither mature nor well-supported.

This leads to the discussion on whether applications like this should run the core algorithms locally or on a server. A server-based architecture has the advantage that more complex recommendation algorithms can be run, efficiently combining similar queries from many users. This makes the mobile app simply a user interface, with all of the major computation being done on the server, allowing also older and slower models to run the app. Furthermore, with all queries and user requests processed at the server, it is easy to gather more data, e.g., which objects are visited most often and at what times, etc. However, this comes at the cost of having to maintain a server that has enough computational power. For small companies or individuals, it may be prohibitive working with Linked Open Data as it represents a barrier to entry.

On the other hand, one could opt to do all computations on mobile device, alleviating the need for a central server. Since this is a cost-efficient way to work with Linked Open Data, it is available for both large parties and individuals, facilitating the dissemination of Linked Open Data. However, as of now, there is a lack of mature and stable APIs for working with Linked Open Data on mobile platforms such as Android. It takes a lot of developer efforts to get all the required libraries running. On a short term, this might be the most prohibitive factor of working with Linked Open Data directly on mobile devices. Last, since mobile devices have limited computing power and battery life, recommendation algorithms are limited to relatively simple ones (i.e., running in P-time).

6 Conclusion

Querying Linked Data endpoints from mobile devices can be relevant for several applications but requires maturing the existing APIs and a larger community effort to make SPARQL as competitive as other web services protocols. Also the data regularity is important to build fluid end-to-end applications. A super cultural heritage ontology would be beneficial to abstract over the specificities of particular datasets by linking, for instance, museum-specific representations to a more standardized way of representing artifacts and exhibitions.

For the future work we consider implementing our algorithm on a backend server to (i) estimate the gain/loss in execution time and (ii) extend the recommendation algorithm with more advanced features, .e.g., semantic similarities based on the artifacts RDF types and the structure of the underlying RDFS schema. Also, with a backend server, it would be interesting to track the number of visitors per room/artifact and use this information in the recommendation process. Among the mid-term perspectives we consider computing tour patterns using semantic similarities or usage mining.

References

1. Androjena: Jena Android porting. `https://code.google.com/p/androjena/` (2010), [Online]
2. ARQoid: Porting of Jena's ARQ to the Android platform. `https://code.google.com/p/androjena/wiki/ARQoid` (2010), [Online; Updated Sep 19, 2010]
3. Bray, T.: The javascript object notation (json) data interchange format (2014)
4. Crofts, N., Doerr, M., Gill, T., Stead, S., Stiff, M.: Definition of the cidoc conceptual reference model. ICOM/CIDOC Documentation Standards Group. CIDOC CRM Special Interest Group 5 (2008)
5. Miles, A., Bechhofer, S.: Skos simple knowledge organization system reference. W3C recommendation 18, W3C (2009)