CURIOS Mobile: Linked Data Exploitation for Tourist Mobile Apps in Rural Areas

Hai H. Nguyen, David Beel, Gemma Webster, Chris Mellish, Jeff Z. Pan, and Claire Wallace

{hai.nguyen,d.e.beel,gwebster,c.mellish,jeff.z.pan,claire.wallace}@abdn.ac.uk
dot.rural Digital Economy Hub
University of Aberdeen, UK

Abstract As mobile devices proliferate and their computational power has increased rapidly over recent years, mobile applications have become a popular choice for visitors to enhance their travelling experience. However, most tourist mobile apps currently use narratives generated specifically for the app and often require a reliable Internet connection to download data from the cloud. These requirements are difficult to achieve in rural settings where many interesting cultural heritage sites are located. Although Linked Data has become a very popular format to preserve historical and cultural archives, it has not been applied to a great extent in tourist sector. In this paper we describe an approach to using Linked Data technology for enhancing visitors' experience in rural settings. In particular, we present CURIOS Mobile, the implementation of our approach and an initial evaluation from a case study conducted in the Western Isles of Scotland.

1 Introduction

From the data perspective, most heritage based mobile applications (apps) to date have used content specifically tailored for the apps. This content usually follows a pre-set geographical route and story-lines while requiring a large amount of human effort to generate. Even though this approach can provide user-friendly and concise content to visitors, it is not very practical for small local community groups. Local community groups often develop their cultural heritage collections in an archive instead of a collection of stories that can be presented to visitors. Moreover, if this data covers a large, non-linear geographical area, it is not realistic to use pre-set geographical routes and expect visitors to follow them.

Linked Open Data is a set of best practices to publish structural data on the web, as introduced by Tim Berners-Lee [1]. It has several important advantages over traditional relational databases such as integrability and reusability, which has made it become increasingly popular within the cultural heritage sector. There have been several efforts to bring cultural heritage archives into Linked Data formats such as in the CultureSampo project [2] and the OpenART [3] project. The ultimate goal of such projects is to allow data to be able to contextualised, reused, and integrated further.

The work described in this paper was carried out as parts of the CURIOS Mobile project at the University of Aberdeen, a follow-up of the CURIOS project.[1] The

[1] http://curiosproject.abdn.ac.uk

CURIOS project aims to produce a set of software tools to allow small local community groups to produce and consume their cultural heritage archives in Linked Open Data formats. The main software is a Linked Data Content Management System (CMS) [4,5].This Linked Data CMS provides a platform for novice users to produce more Linked Data by providing a friendly user interface as in the traditional CMSs such as Wordpress or Mediawiki.

Given the rich content generated in Linked Data formats, CURIOS Mobile explored the ways to exploit Linked Data in order to provide visitors with an enjoyable user experience while exploring rural areas. The objective of CURIOS Mobile was to deliver a tourist mobile application which can 1) exploit the current linked dataset generated by the CURIOS CMS and 2) work reasonably well with unreliable mobile Internet connection in rural areas. The Hebridean Connections' dataset has been used as the main case study for the CURIOS Mobile project.

There have been several attempts to bring Linked Data and the Semantic Web closer to mobile devices: DBPedia Mobile [6], mSpace Mobile [7], *Who's Who* [8], etc. However, previous work either focuses on only specific problems such as context discovery and visualisation, or assumes that data connection is always available and reliable. In contrast, in this paper we present a generic framework to exploit Linked Data archives for tourist activities, especially in a rural context where limited or no data connection is available to mobile devices. Below are the summary of the paper's main contributions.

1. A generic framework to use Linked Data archives for tourist activities via a mobile application.
2. Linked Data-based caching solutions to the unreliable data connection issue in rural areas.
3. A recommendation mechanism to choose which information to present to visitors on the site based on data characteristics.
4. A simple mechanism to generate text-based descriptions directly from RDF triples based on the techniques introduced in [9].

This paper is organised as follows. In Section 2 we briefly introduce the context of this paper; in particular the technologies and dataset we are using, as well as the broader framework as to where the system described in this paper fits. In Section 3 we introduce the main challenges for developing and deploying tourist mobile applications in rural areas. In Section 4 we introduce our approach, including a brief overview of the system and our solutions to the challenges mentioned in Section 3. In Section 5 we present some preliminary results of our implementation. Section 6 is the discussion of related work. Section 7 includes the conclusion and some pointers to future work.

2 Background

2.1 Linked Open Data

A 4-star Linked Open Data as described in Tim Berners-Lee's note [1] would use HTTP URIs[2] to denote things (i.e., individuals) and W3C standards such as RDF or

[2] Uniform Resource Identifier

OWL[3] to describe such individuals' information or to relate one individual to another. The Resource Description Framework (RDF) is used as a standard format to describe things and their relationships within a linked dataset as RDF triples. The RDF triples are then stored in a type of database system, namely a triplestore and can be retrieved or maintained via a specific query language, SPARQL[4]. The vocabularies used to describe things and their relationships using RDF are usually defined in an OWL ontology.

2.2 The CURIOS Project

The CURIOS project aims to provide a sustainable and extensible software system for historical societies to produce and consume cultural heritage data in the form of Linked Open Data. By combining Linked Data standards and software with Drupal, a popular open source CMS, CURIOS provides users with limited knowledge on semantic technology a friendly front-end in order to produce linked data without noticing the underlying technologies (e.g., SPARQL, RDF). In CURIOS, the data entered by users are stored in a triplestore while the configuration of how data are presented to users is stored in Drupal's traditional SQL database. This approach allows the linked dataset maintained by CURIOS to be loosely coupled to Drupal meaning it can be reused in different applications or by other software. There have been two main case studies conducted to evaluate the CURIOS system: one involving historical societies based in the Western Isles of Scotland (Hebridean Connections) and another one with a local historical group at Portsoy, a fishing village located in the North East of Scotland. These two case studies are very different in terms of dataset's scale and the organisation structure. However, CURIOS has been well-received from both communities.

2.3 The Hebridean Connections Case Study

Hebridean Connections is a project connecting local historical societies across the Western Isles of Scotland (a.k.a. Outer Hebrides). Thousands of records about the genealogy, places, traditions, cultural and history of the islands have been generated by local historical societies and their contributors. Before the release of CURIOS, the data had been preserved in multiple physical archives by each local historical society and in a relational database maintained via proprietary software. This dataset has been digitised and preserved in a Linked Open Data standard format (RDF). More and more linked data has been added into the archive using the CURIOS Linked Data Content Management System since its first release (February 2014). The CURIOS system has been deployed on the Hebridean Connections website recently (available at `http://www.hebrideanconnections.com`).

As of 28/07/2014, the dataset in the Hebridean Connections case study consists of 864,429 RDF triples before inference, incorporated within a relatively simple OWL ontology. These triples form a total of 44,358 records. Basically, a CURIOS record is the set of triples, usually presented together, describing a particular subject (identified by a URI). Formally, a CURIOS record is defined as follows.

[3] Web Ontology Language
[4] Simple Protocol And RDF Query Language

Definition 1 (CURIOS Record) *A CURIOS record r_s of a subject s is a set of RDF triples of the form $< s, p, o >$ where s is the URI identifying the record by its numeric identifier, p is either a datatype property or an object property, and o is either a literal value or a URI.*

Figure 1 shows an example of a person record within the Herbidean Connections' dataset. Each record in the dataset will have at least a subject ID(e.g., `23160`), a title (e.g., `"Angus Macleod"`) and a record type associated with it (e.g., `hc:Person`). Moreover, depending on its type (e.g., `hc:Person`, `hc:Location`, etc.), a record can have different datatype properties. For example, only `hc:Person` records can have `hc:occupation` datatype property while `hc:Location` records can have geographical information (i.e., `hc:easting` and `hc:northing`). A record might also contain "links" (i.e., object properties) to other records such as `hc:childOf`.

```
hc:23160 hc:title "Angus Macleod"
hc:23160 hc:subjectID 23160
hc:23160 rdf:type hc:Person
hc:23160 hc:sex "Male"
hc:23160 hc:description "Angus Macleod was born in 1916 to 8 Calbost..."
hc:23160 hc:approvedForPub "yes"
hc:23160 hc:bkReference "CEP 2335"
hc:23160 hc:isChildOf hc:23112
hc:23160 hc:isBornAt hc:369
```

Figure 1: Part of a record of a person in Hebridean Connections' dataset.

Figure 2 shows some statistics of the Hebridean Connections dataset (as of 28/07/2014), including the total number of records grouped by categories, published records, records with geographical information and records with a description. Note that even though the records have already been digitised and stored in a triplestore, not all of them are available to the public for browsing because parts of the datasets are still under revision. The records are grouped into 17 categories and records about people and places make up a large proportion of the dataset. Column **#geog info** shows the number of records with geographical information (e.g., having `hc:easting` and `hc:northing` properties to represent an Ordnance Survey grid reference). Most records also have a description, a human-generated text giving more information about the record. Some of the description might have annotations, i.e., links to other records. However, not all records have a description, as shown in the last column of Figure 2.

3 Challenges to Tourist Mobile Apps in Rural Areas

3.1 Unreliable Connectivity and Expensive Download costs

Tourist mobile applications can be grouped into two main groups: *on-the-fly download* and *one-off download* approaches. The on-the-fly download approach provides the most

Record types	**#records**	**#published**	**#geog info**	**#with description**
Vehicles	30	6		29
Sound files	39	39		39
Gaelic verses	55	51		55
Historical events	81	71		79
Businesses	82	65	4	69
Natural landscape features	92	64	62	77
Organisations	123	103		111
Resources	190	88		146
Buildings and public amenity	206	122	75	142
Objects and artefacts	216	215		215
Stories, reports and traditions	448	435		446
Boats	487	412		450
Locations	1,137	700	204	757
Landmarks and archaeological sites	2,255	2,221	2,198	2,243
Croft and Residences	2,818	1,527	498	1,368
Image Files	3,226	3,185		3,065
People	32,873	17,398		27,883
Total	44,358	26,702	3,041	37,174

Figure 2: The Hebridean Connections dataset

up-to-date information and requires much less initial download. A typical category of applications following this approach is the so-called mobile web-apps. Every time a user requests some information, the application pulls the response onto the device and displays it to the user. This information might or might not be stored on the device. An advantage of this approach is that users generally only download (pay for) what they browse. This approach is best suited for central places of interests such as museums and galleries where mobile Internet connection is generally good or where Wi-Fi connection is available. Unfortunately, in rural areas the apps using this approach will become very unresponsive, or in the worst case, will not work at all. For example, when there is no Internet connection, the Brighton Museum app[5] only prompts a message that users need to have Internet connection to simply open the app.

The one-off download approach, in contrast, relies on an assumption that the visitors intentionally use the mobile app for their exploration activities. A typical scenario is that the visitors download the app at home or at places with good Wi-Fi connection and then bring the app with them on-site for offline usages. This approach therefore requires a heavy initial download in order to ensure good user experience in terms of rich contents (more audio, images and videos) and a responsive user interface (all data is kept on the device). Some examples using this approach are the Timespan - Museum Without Walls iOS app[6] and Great Escape Moray published by the National

[5] `https://play.google.com/store/apps/details?id=com.surfaceimpression.brightonmuseums`

[6] `https://itunes.apple.com/gb/app/museum-without-walls-scotlands/id556429487`

Library of Scotland[7]. These apps require an initial download from 350MB to more than 450MB. Despite bringing rich and layered content to users, the one-off download approach suffers several drawbacks, especially in a rural context. Firstly, visitors need to know about the app before coming to the site. This is suitable for popular tourist sites but not for the small-scaled, remote areas less known for tourist activities. Secondly, as the data is downloaded as a whole, it is not as pertinent for tourists who only have certain interests in the information. For example, the user might only be interested in certain places or people only, and in this instance it would be best for the app to only pre-download such information. Thirdly, applications using this approach cannot cope with frequently updated data. For this approach, updates are done much less often and usually require another heavy data download.

3.2 Manual Recommendation in a Large Archive

Given the large total number of records in the archive (about 45 thousands records in which over 3000 are places), it is challenging to choose which ones should be presented to the users. A simple solution is to ask for recommendations from local people. Hebridean Connections, our local partner, helped to recommend 325 records related to the Pairc area of Lewis, Scotland. Of these 325, 64 records are about places of interests but only 55 of these are available to the public due to on-going revisions. Clearly, recommending 325 records (64 places) over 45 thousand records (3000 places) is not a trivial task and requires much time and effort.

In addition to this, even when there is a list of suggested records from local people, it is also not easy to choose which records should be given to the users and/or should be cached in the users mobiles. For example, some records are very general and can be related to many other records such as World War I, Pairc Historical Society. For example, a major event such as “World War I” can be associated with hundreds of other records ranging from people to places and stories. However, a user viewing the “World War I” record is not necessarily interested in all several hundreds of related records. In fact, presenting all records would be not only expensive in terms of downloading cost and time, especially in the rural context, but also confusing to users, as there would be too many records to browse.

3.3 Presenting Records without a Description

For such a large, crowd-sourced archive like Hebridean Connections, it is often the case that not all records have a detailed description. Recall from Section 2.3, Figure 2 shows that only 83.8% (37,174 out of 44,358) of the records have a description. This might not be a very important issue if the data are only browsed on the web or shared in a linked data cloud. However, for tourist applications, the lack of detailed, human-readable description will significantly affect the experience of visitors. Fortunately, despite the lack of a human-generated description, these records still have a set of triples with datatype

[7] `http://www.nls.uk/learning-zone/great-escapes`

properties and object properties, which can be used to generate a very simple description. Therefore, instead of presenting to visitors a set of raw RDF triples and no description, the system should be able to generate a simple description based on the RDF data of that record.

4 The CURIOS Mobile System

4.1 System Architecture

The CURIOS Mobile System adopts the client-server software model, as shown in Figure 3. As can be seen, CURIOS Mobile is a component of the whole CURIOS System,

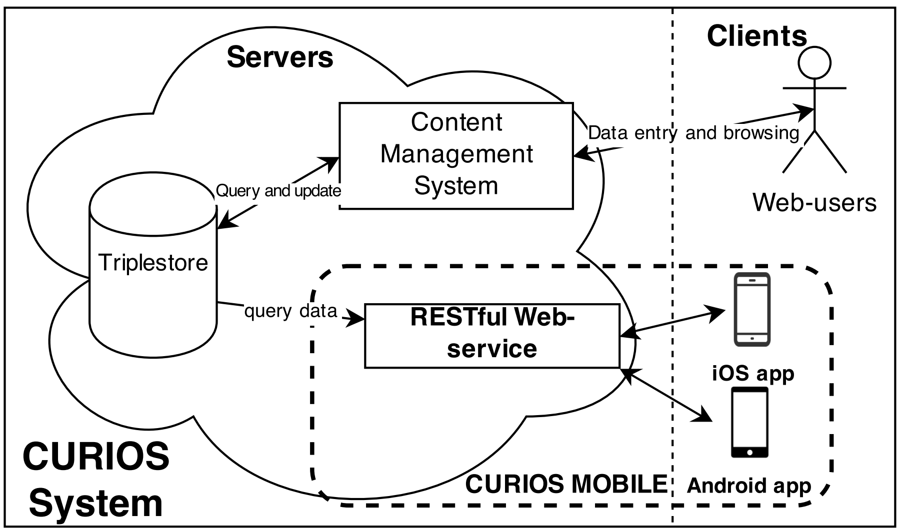

Figure 3: CURIOS Mobile System

in which it reuses the database (the triple store) maintained via the CURIOS CMS. This database is wrapped by a RESTful[8] web service, which provide data to the mobile devices via an API (Application Programming Interface). By using this software model, the same API/Web-services can be used for multiple mobile platforms: Android, iOS, Windows Phone, etc. However, unlike the traditional client-server software model, where the client side is very thin and only shows results retrieved from the server, the client side in CURIOS Mobile system also caches and stores data in a database. Below we describe briefly the components of the CURIOS Mobile system as well as the main CURIOS system and how they interact to each other.

[8] REpresentational State Transfer (REST) is a web architecture style commonly used for the implementation of web-based APIs [10].

The CMS is not a component of the CURIOS Mobile system in particular, but a key component for the whole CURIOS system, where it allows users (members of historical societies) to enter and curate data. It is also the central place for the public to browse and explore the archive as well as to contribute to the dataset in some way via social media tools such as the commenting system. The CMS uses both a relational database as a back-end for Drupal's internal data (e.g., nodes, entities) and a triplestore to store data (records) generated from users.

The triplestore is used for both the CMS and the CURIOS Mobile system. The RDF triples are stored in the Jena TDB persistent storage system. As the same triplestore serves multiple applications (e.g., the CMS and the RESTful web service), Jena Fuseki SPARQL server [11] is used for data retrieval and maintenance. Data held in the triplestore are generated and maintained via updates sent from the CMS. In return, the triplestore answers the requests sent from the CMS and the RESTful web service and hence allow users or client apps to browse the dataset.

In theory, the advantage of Linked Data technologies is to enable open data, and hence the triplestore can be accessed (read-only) by web-users directly without going through the middle-services such as the CMS or the RESTful web service. Therefore, the CURIOS Mobile system or the CMS are just examples of how the linked datasets can be used and reused. In the CURIOS Mobile project, there are two important reasons why direct access to the triplestore should be avoided: access control and data pre-processing. Recall from Section 2.3, parts of the dataset are not visible to the public and some datasets, e.g., in the Hebridean Connections case study, require a complex hierarchy of user-roles and associated permissions, which needs to be implemented in a middleware like the CMS. Another reason is that some of the data are not yet ready to be presented in their current form and hence require some pre-processing before sending back to the client apps/end-users. For example, the coordinate system used in the Hebridean Connections' dataset is the Ordnance Survey Grid Reference (e.g., using easting/northing) while most mapping API/services for mobile devices nowadays use the latitude/longitude system.

The RESTful web service provides an easy, flexible way for mobile devices to request and retrieve data from the server.[9] Beside the main role as an API for mobile clients to access the data stored in the triplestore, this sub-component of CURIOS Mobile also has three more tasks: caching recommended data from the triplestore; generating narratives for records without a description; and pre-processing data before sending to the client apps under JSON format. The web service is also responsible for validating/updating cached entries, e.g., to see if a record has been update recently, and send corresponding updates to the client apps so that the second layer of data caching (see the next section) is kept up-to-date.

[9] The RESTful web-service is currently hosted at `http://curiosmobile.abdn.ac.uk:8080/CuriosMobile` and the API Documentation is given at `http://docs.curiosmobile.apiary.io`.

The client apps include a GUI (Graphical User Interface) for end-users to browse the dataset and a second layer of data caching. More specifically, data retrieved from the RESTful web service will be stored in a SQLite database of the device. This caching layer would optimise the performance of data browsing using the mobile devices, especially with unreliable Internet connectivity. Three main tasks of the client apps are:

– to allow users to view the archived records using the mobile devices,
– to cache records in advance based on users' location and the previous browsing history, and
– to send notifications about places of interest when they approach them.

4.2 Data Caching for Rural Settings

As mentioned above, CURIOS Mobile maintains two layer of data caching, one in the RESTful web service and one in the client apps. The first layer is to avoid overhead while querying data against the triplestore as well as to keep track of which records have been out of date. In this section we focus on the second layer of data caching since this is used to overcome the problem of unreliable Internet connection in rural areas.

The caching services can be used to download relevant data when the visitors have good Internet connection such as WIFI or 3G. Relevant data will already be downloaded and stored in client apps even before the visitor arrives at the places of interest. When the visitor goes to rural areas with limited or even no access to the Internet, the app will still be able to use seamlessly. Below we present two caching services implemented in CURIOS Mobile: *Location-based Caching* and *Semantics-based Caching*.

Location-based Caching is to cache the records with geographical information (i.e., a pair of easting/northing or latitude/longitude) based on the current user's location. Therefore, only the records located around the geographical area the user is visiting are cached. To adapt different case studies and different geographical areas, we use a parameter, the *euclidean distance* (denoted by $distance_e$) between the user's current location and the record's location, to adjust the level of of caching. Records within the radius of $distance_e$ from the user's current location are cached. However, one should be careful while adjusting $distance_e$ as this parameter should be proportional to the average distance between places of interest to be most effective. For example, if $distance_e$ is much greater than the average distance between places, most places will be cached, and the client apps end up downloading all the dataset even when the visitors are not keen on viewing all the places. In contrast, if $distance_e$ is smaller compared to the range where mobile data connection (signal) is available, it is likely that no record can be downloaded and cached due to no Internet connection.

Semantics-based Caching As the data used within CURIOS Mobile is represented as linked data, it is also possible to perform ahead caching for the related records with respect to what the users have been viewing. We call this style of caching *Semantics-based Caching*. Similar to Location-based Caching, we use a parameterisable distance to control the level of caching needed. We refer to this distance as the *semantic distance*, as defined in Definition 2.

Definition 2 (Semantic Distance) *The semantic distance of two CURIOS records r_{s1} and r_{s2} in a triplestore R, denoted by $distance_s(r_{s1}, r_{s2})$, is the length of the shortest path in the RDF graph connecting $s1$ and $s2$.*

If two records are directly linked, i.e., $< s1, p, s2 >\in R$ then $distance_s(r_{s1}, r_{s2}) = 1$. However, like location-based caching, it is pertinent that this distance should be chosen carefully to avoid downloading too much data, which might not be relevant enough to users. For example, CURIOS Mobile only downloads and caches records one link away from the viewing record, i.e., $distance_s(r_{s1}, r_{s2}) = 1$. These records will be downloaded in the background process and hence cannot interfere with users current activities. Subject to successful downloads, these records are stored in the app's SQLite database. In reality, some records are general and linked to many other records, and it is not practical to download and store all of them. Therefore, it is necessary to have a mechanism to sort the records based on the level of interest, and then only pick the top of them (i.e., the most interesting ones). This mechanism will be described in detail in the following section.

4.3 Auto-recommendation of Things of Interest

As presented in Section 3.2, it is not trivial to manually choose which records should be presented to users in a large archive like the Hebridean Connections's dataset. To tackle this challenge, we have constructed a *utility function* to assess the level of interest in each record. Firstly, if a user is interested in some records or contents by viewing or searching for them, not only these records but also the related records (selected via the semantic distance in Definition 2) can be the potential candidates for caching. Secondly, to avoid over-caching uninteresting records, related records will be sorted, picked, downloaded and cached based on their level of interest (the result of the utility function).

While assessing the level of interest of a record, there are three factors taken into account: 1) the recommendations from local people, 2) the quality of the record description (based on text length) and 3) the number of links to and from that record, as described in the below equation:

$$Utility(r) = p_1 \times is\_suggested(r) + p_2 \times description\_quality(r) + p_3 \times links\_quality(r)$$

where r is a CURIOS record, $is\_suggested(r)$ is a binary bit representing whether the record has been suggested by locals, $description\_quality(r)$ ranging from 0 to 1 represents the quality of r in terms of its description and $links\_quality(r)$ ranging from 0 to 1 represents the quality of the record r in terms of the links from/to it, and p_1, p_2, p_3 the preferences given to each factor of the utility function.

Note that the utility function can be easily modified to adapt different datasets/case studies. For example, in CURIOS Mobile we set p_1, p_2 and p_3 the values of 0.5, 0.3 and 0.2 respectively so that the priority is then given to suggested records and records with higher quality description. For example, a record without a description and which is not suggested by locals will unlikely be selected and cached. Links to and from that record are also taken into account, but with a lower preference. This factor only counts

20% towards the final utility value, because this might add noise to the utility function, particularly in cases where the record is a generic term such as a book, a historical society or an important historical event (e.g., "World War I").

The utility value of a record can be used in a couple of ways. Firstly, the app can have a threshold value to allow some records to be downloaded and cached in the clients apps based on the utility values. This initial set of cached records will be used to give basic information to the user in the beginning. After that, as soon as the user starts to browse this set of records, other records related to this set will also be selected and cached using the semantic distance. Eventually the set of cached records will be growing until there is no Internet connectivity or no user activity. Secondly, when a record is linked to many other records, it is possible to compare the utility values of these linked records to decide which ones should be downloaded and presented to users. In CURIOS Mobile, these tasks are done on the server side, before sending back the record to mobile clients.

4.4 Description Generation from RDFs

To present records without a description to users (see Section 3.3), we adopted the *free generation* approach similar to the one in the Triple-Text system [9]. An example of a CURIOS record which only contain RDF triples and no description is shown in Figure 4. Now the general ideas are that each text sentence is generated from an RDF triple and the verbs in these sentences are constructed directly from RDF property names. An advantage of this approach is that it is domain-independent, meaning that this can be used as a generic approach for CURIOS Mobile to exploit any archived linked data generated via the CURIOS system. However, this approach requires the ontology designers to think ahead of properties and concept names so that they can be used later to generate correct verbs and nouns. Narrative generation is done on the server side of CURIOS Mobile (i.e., in the RESTful web service) using the SimpleNLG library [12].

The description generation process for a record has three steps. Firstly, all RDF triples within the record containing irrelevant data such as metadata (e.g., record owner, publication status) are filtered out. Secondly, we build an abstract model of a record from the RDF triples. A record has a title, a type, a set of datatype properties and their values (e.g., `hc:occupation:Teacher`) and a set of object properties (e.g., `hc:isChildOf`) and their values' titles. The names of datatype properties and object properties are usually nouns and verbs respectively. In Figure 4, although `hc:dateOfBirth` and `hc:dateOfDeath` are listed as datatype properties, they are in fact object properties linking to a `hc:DateRange` individual. The values of the `hc:DateRange` individual is computed from a a pair of time points. A pair of time points can cover different periods, e.g., a date, a year, a decade, a century, etc. This approach is to deal with inexact dates that occur very frequently in the cultural heritage domain. This also explains why in the generated text there are different formats of "date" (see Figure 5). Thirdly, we generate a description for the record based on this abstract model, sentence by sentence. The generated description of a record is a paragraph consisting of: a sentence describing the record's title and type, a set of sentences generated from the datatype properties, and a set of sentences generated from the object properties. Because datatype properties are usually named as nouns, we used possessive adjectives

```
hc:10738 hc:ownedBySociety "CEBL"
hc:10738 hc:approvedForPub "yes"
hc:10738 rdf:type hc:Person
hc:10738 hc:subjectID 10738
hc:10738 hc:title "Margaret Smith"
// datatype properties
hc:10738 hc:sex "Female"
hc:10738 hc:BKReference "CEBL 73"
hc:10738 hc:alsoKnownAs "Mairead Iain Mhoireach"
hc:10738 hc:dateOfBirth hc:Dr6411
hc:10738 hc:dateOfDeath hc:Dr641
// object properties
hc:10738 hc:married hc:10708
hc:10738 hc:livedAt hc:792
hc:10738 hc:livedAt hc:780
hc:10738 hc:isChildOf hc:861
hc:10738 hc:isChildOf hc:863
hc:10738 hc:isParentOf  hc:5521
hc:10738 hc:isParentOf hc:6015
hc:10738 hc:isParentOf hc:4219
hc:10738 hc:isParentOf hc:17394
hc:10738 hc:informationObtainedFrom hc:5181
```

Figure 4: RDF triples of the record `hc:10738`

(e.g., his, her, its) for sentence construction. Similarly, we use pronouns (e.g., he, she, it) for sentences constructed from object properties. All generated sentences are added into a paragraph and realised by SimpleNLG.

```
Margaret Smith also known as Mairead Iain Mhoireach is a person.
Her date of death is Thu, 18 Dec 1873. Her date of birth is 1801.
Her bk reference is CEBL 73. Her sex is Female. Margaret Smith is
parent of John Gillies, Henrietta Gillies, Christina Gillies and
Peter Gillies. She is child of John Smith and Catherine Maciver.
She marries Angus Gillies. She lives at Bosta and 1 Earshader. She
informations obtained from Register of Births, Marriages and Death.
```

Figure 5: The generated description for `hc:10738` from the triples in Figure 4

Figure 5 shows the description generated from the RDF triples in Figure 4. Datatype properties which are not recognised as a noun such as `hc:alsoKnownAs` are used as complements for the subject in the first sentence. For object properties with multiple values such as `hc:isParentOf` and `hc:isChildOf`, instead of producing multiple similar sentences which cause redundancy and disinterest, the RDF triples are aggregated into one sentence. If there are too many values for an object properties (e.g., more than 5), the system only generates the total count and some records, sorted by the level of interest using the utility function mentioned in Section 4.3. It can be seen that this approach cannot deal with object properties whose names are not verbs. For example, `hc:informationObtainedFrom` is translated as a verb "informations obtained from". Also, the articles such as "a/an/the" are also missing.

5 Preliminary Results

In this section we present some preliminary evaluation of our implementation of the data caching services. The testing were done at the Pairc area in the Isle of Lewis, Scotland. As shown in Figure 6, three Android-based devices with different settings are used to evaluate the caching services.

#Device	Network	Device	OS	Caching distance	Test
Device 1	3 UK	Moto G (3G)	Android 4.4.2	12km (3G available)	Both
Device 2	O2 UK	Moto G (3G)	Android 4.4.2	7km (only GPRS available)	Location-based caching
Device 3	No 3G	Nexus 7	Android 4.4.3	Not applicable	Semantics-based caching

Figure 6: Devices used for testing

We aimed to use Device 1 for testing the combination of both caching services, Device 2 for testing location-based caching only, and Device 3 for testing semantics-based caching only. Device 1 and 2 were able to use data connection (e.g., 3G or GPRS) while Device 3 could only access to WIFI connection and hence Device 3 could not use the location-based caching during the journey. Because 3G data connection is only available within 10-12km of the site and only GPRS is available within 5-7km of the sites, to test the *location-based caching* service, we adjusted the caching distance ($distance_e$ in Section 4.2) to 12km and 7km for Device 1 and 2 respectively. By doing so, Device 1 could use the 3G network for its location-based caching service while Device 2 could only use the slower GPRS for its caching service. Information about how strong the signal would be in different areas along the route can be obtained easily from any operator's online coverage map.

To test the *semantics-based caching* service, before going to the site, users for Device 1 and Device 3 were asked to randomly browse and read records for 15 to 20 minutes. After browsing, the number of cached records for Device 1 and Device 3 were 230 and 128 respectively. The number of cached records in Device 2 remained 0.

During the journey towards the site, Device 1, with a 12km caching distance setting, had increased the number of cached records from 230 to 285 and settled at this number (all 55 recommended places had been cached) at about 10km away from the Pairc area. The user with Device 1, with 285 cached records, could browse most records about places and related things such as people, stories, etc. on site even without an Internet connection. Only 15% of the clicked record could not be rendered, meaning that 85% of clicked records are cached by both caching services.

Device 2 got some messages showing that some of the to-be-cached records had not been downloaded successfully while driving at a speed of approximately 80-90km per hour. When stopped, however, after remaining in a location for several minutes the caching resumed correctly. This is because Device 2 could only use GPRS data connection which is much slower than 3G meaning that the downloads were interrupted while moving fast between places with and without a data connection. At 6km away from the sites, 16/55 records had been cached and when arriving at Ravenspoint, a place within the Pairc area and about 4-5km away from the nearest suggested place, all 55/55 records had been cached in Device 2. This suggests that the location-based caching service worked well, even with a slow data connection such as GPRS.

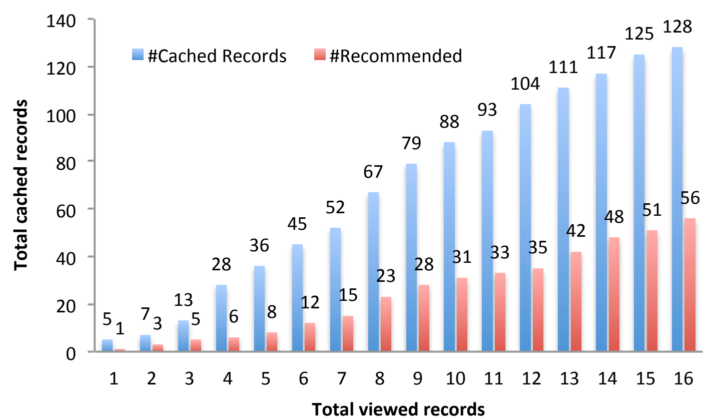

Figure 7: The cache's growth using the semantics-based caching service.

For Device 3 (only using *semantics-based caching*), the user was asked to browse the places which they would like to visit on the map as well as the records related to these places randomly. We measured the number of records which the user clicked to view, the number of cached records after viewing the record, and how many of the cached records were recommended by locals. Figure 7 shows the growth of the cache while the records were browsed randomly. Firstly, the cache size (blue column) almost grows linearly to the number of viewed records. This means that the system did not over-cache records and hence could keep the cost of data download low for visitors. Obviously, as a visitor approaches to browsing all records then the cache will settle at some point because there would be nothing left to cache. Secondly, in average, recommended records (red column) count about 35% of the cache, meaning that the semantics-based caching service downloaded a good amount of high quality records, which would be very likely to be viewed by visitors when they arrived at the site. Like location-based caching, records cached using semantics-based caching can be used to send pushed notifications to users while they are passing the places related to the cached records. This process can be done completely offline as only user's current location is required. Therefore, users would be able to read records along the way, even in the areas with no connectivity.

The initial results suggested that although the combination of both caching services worked best, it would still be possible for devices without 3G service like Device 3 to use semantics-based caching to give users information about interesting places (and related records) as they are passing them. As for location-based caching, the results showed that not only 3G but also GPRS data service was sufficient for downloading the records. However, the coverage pattern of Internet availability and the user's moving speed should be taken into account while performing location-based caching.

6 Related Work

There have been a significant amount of work on the Semantic Mobile Web. DBPedia Mobile [6] is a location-aware mobile app which can explore and visualise DBPedia resources around the user's location. DBPedia Mobile also maintains a cache of resources on the server side similar to the first layer of caching in CURIOS Mobile. mSpace Mobile [7] is a framework that can retrieve Semantic Web information (RDF) from different sources and visualise the retrieved data to users. However, unlike our approach, in both approaches mentioned above there is no or only minor caching on the client side. These approaches require a reliable Internet connection to operate and hence would be difficult to work in a rural context. In terms of caching services, the most relevant work to our approach is *Who's Who* [8] althought this work does not allows location-based caching. In *Who's Who*, data retrieved from the server side are kept in a local RDF store in the devices and SPARQL queries are used to retrieve requested information from the device's RDF store.When needed, the device will ask for more triples from the server. The main difference between our caching service and *Who's who* is that in the mobile device's caching layer, we use the mobile SQLite database instead of an RDF store. This solution is much simpler for deployment as no RDF store needs to be installed and operated in the device while a good performance can still be achieved. Another difference is that our approaches does use the semantics-based caching service for caching related data in addition to the requested data.

Previous works on natural language generation from linked data mostly focus on using the ontology statements to generate text, such as NaturalOWL [13] or SWAT [14]. These approaches therefore require taking the ontology as an input to generate text. However, this is not possible in CURIOS Mobile, where the server side has access to only the triple store via a SPARQL server but not the OWL ontology. As a result, it is necessary that the description of a record must be generated directly from RDF triples. Therefore we adopted the approach introduced in [9]. Although it was not perfect (e.g., grammatical errors and ontology design commitments), we believe that it is a simple yet acceptable solution to the problem of presenting records without a description.

7 Conclusion and Future Work

In this paper we described CURIOS Mobile, a framework to exploit linked data-based archive for tourist activities in rural areas. We showed how our approach can overcome the connectivity issues in rural areas by using different caching services, especially the semantics-based caching which takes advantage of the Linked Data format. In addition, we introduced the mechanisms to recommend things of interest in a large archive such as the Hebridean Connection dataset. Finally, we presented a simple mechanism to generate text from RDF triples for records without a description.

There are two important directions of future work. Firstly, the current CURIOS Mobile system only focuses on the consumption of linked data and hence it would be interesting to explore how to extend the system to allow the production of linked data from visitors. Secondly, we would like to investigate how to improve the utility function and the narrative generation service so that visitors' preferences and activity history can be taken into account.

Acknowledgements The research described here is funded by the SICSA Smart Tourism programme and the award made by the RCUK Digital Economy programme to the dot.rural Digital Economy Hub (award reference: EP/G066051/1). We would like to thank our project partners, Hebridean Connections and Bluemungus, for their comments and support.

References

1. Berners-Lee, T.: Linked-data design issues. W3C design issue document (June 2009) http://www.w3.org/DesignIssue/LinkedData.html.
2. Mäkelä, E., Hyvönen, E., Ruotsalo, T.: How to deal with massively heterogeneous cultural heritage data – lessons learned in CultureSampo. Semantic Web **3**(1) (2012) 85–109
3. Allinson, J.: Openart: Open metadata for art research at the tate. Bulletin of the American Society for Information Science and Technology **38**(3) (2012) 43–48
4. Taylor, S., Jekjantuk, N., Mellish, C., Pan, J.Z.: Reasoning driven configuration of linked data content management systems. In: JIST 2013. (2013)
5. Nguyen, H.H., Taylor, S., Webster, G., Jekjantuk, N., Mellish, C., Pan, J.Z., Rheinallt, T.A.: CURIOS: Web-based Presentation and Management of Linked Datasets. In: Proceedings of the ISWC 2014 Posters & Demos Track. (2014)
6. Becker, C., Bizer, C.: Exploring the geospatial semantic web with dbpedia mobile. Journal of Web Semantics **7**(4) (2009) 278–286
7. Wilson, M.L., Russell, A., Smith, D.A., Owens, A., m.c. schraefel: mSpace Mobile: A Mobile Application for the Semantic Web. In: End User Semantic Web Workshop. (2005)
8. Cano, A., Dadzie, A.S., Hartmann, M.: Who's Who – A Linked Data Visualisation Tool for Mobile Environments. In: The Semanic Web: Research and Applications (ESWC'11). Volume 6644. (2011) 451–455
9. Sun, X., Mellish, C.: An Experiment on "Free Generation" from Single RDF Triples. In: Proceedings of the 11th European Workshop on Natural Language Generation. ENLG '07 (2007) 105–108
10. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis (2000)
11. Seaborne, A.: Fuseki: serving RDF data over HTTP. http://jena.apache.org/documentation/serving_data/ (2011) Accessed: 2012-10-27.
12. Gatt, A., Reiter, E.: Simplenlg: A realisation engine for practical applications. In: Proceedings of the 12th European Workshop on Natural Language Generation. (2009) 90–93
13. Androutsopoulos, I., Lampouras, G., Galanis, D.: Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System. J. Artif. Intell. Res. (JAIR) **48** (2013) 671–715
14. Stevens, R., Malone, J., Williams, S., Power, R., Third, A.: Automating generation of textual class definitions from OWL to English. J. Biomedical Semantics **2** (2011)