CURIOS MOBILE

Linked Data Exploitation for Tourist Mobile Apps in Rural Scotland

Hai Nguyen

hai.nguyen@abdn.ac.uk

dot.rural Digital Economy Hub
University of Aberdeen
THE CURIOS PROJECT

• Facilitate historical societies in maintaining the digital archive of their cultural heritage more sustainably and efficiently using open-source software (Drupal CMS) and open data standards (Linked Open Data).

• Use case: Outer Hebrides
  • 45000 records
  • Rich content: people, boats, historical events, places…
  • Dedicated/active historical societies

• Project website: www.curiosproject.abdn.ac.uk
CURIOS - THE CMS FOR LINKED DATA

• Open-sourced CMS: Drupal as a tool to **directly** exploit and maintain **linked data**

• The **dataset** and the **software** are **loosely-coupling**.

• Live demos:
  - http://curios.abdn.ac.uk
  - http://hebrideanconnections.com

• Code: https://github.com/curiosproject/curios
CURIOS MOBILE

• Motivation
  • Reusability of Linked Data
  • One data - Many views
  • Semantic technologies can help in rural areas (connectivity, isolated contents)

• Aim: a mobile app for tourists in rural area.

• Challenges:
  1. Connectivity problem
  2. Huge archive (with noise)
  3. Text generation from RDFs
CURIOS SYSTEM OVERVIEW

Database

Web servers

Clients

RESTful API

CURIOS MOBILE

Triplestore

Content Management System

iOS app

Android app

Read-only

Read-Write
APPROACHES TO TOURIST MOBILE APPS

1. One-off download
   - Rich-content
   - Not up-to-date content
   - **Heavy download:** 40-500 MB per application (excl. updates)

2. Pay as you go requests
   - Needs **reliable** mobile connectivity
   - Usually presented as a Web-app.
   - **CURIOS Mobile:** pre-cached some data as seeds, growing the cache based on semantical relevance and user locations.
A LINKED DATA SOLUTION

- Caching data based on
  - Euclidean distance (location-based caching)
  - Semantic relevance (semantics-based caching)
  - User Preferences
- Help for Initial downloads
LOCATION-BASED CACHING

• Caching data based on Euclidean distance

• Caching distance depends on signal distance

• **Good** for places where Internet connection is **unreliable** ("on/off" pattern)

• **Bad** for places where the whole area of interests does not have any signal at all (over-caching if caching distance is large)

• **Only caching places. How about other record types (people, traditions, events, stories) ?**
SEMANTICS-BASED CACHING

• Caching data based on **semantic relevance** (object-property links)

• Allow **all record types** to be cached

• The more users browse, the more the cache grows

• If a record is linked to many other records, only cache the **most interesting** ones.

• Cached records can be used to push notifications while the user is moving in a no-connection area. (only GPS is required)
AUTO-RECOMMEND CONTENTS

- Huge archive!!!
  - 45000 individual records
  - 33685 are people records
  - 3001 places

- What to pick?
  - Solutions:
    - Locals’ suggestions: 325 records (64 places)
    - How to automate this process?
    - Build a Utility function based on level of significance/user preferences/locals’ suggestions.
    - Level of significance: based on the links to and links from the record URI
    - Also used to sorted records before caching on devices
TEXT GENERATION FROM RDF

• Some records don’t have descriptions.

• How to generate an interesting story?
  • User preferences (people/places/historical events etc)
  • Type of records: people/place

• Proposed solution
  • Directly generate text from RDF triples using properties and classes’ labels (Sun&Mellish, 2007)
  • Only include “interesting” properties for special types (e.g., liveAt, childOf, etc.)
  • Pros: Simple and flexible to data availability
  • Cons: requires commitments from ontology designers
INITIAL RESULT
SCREENSHOTS - ANDROID

Online map

Offline map

A croft

Prototype
SCRENSHOTS - ANDROID

A record

Links

Pushed Notifications from cached records

Prototype
SCREENSHOTS - IOS

Prototype
CONCLUSION

• CURIOS and CURIOS Mobile

• Linked data-based solutions for
  • Unreliable connection problem in rural areas
  • Auto-recommend interesting contents
  • Generate texts directly from RDF