Merging and Enriching DCAT Feeds to Improve Discoverability of Datasets

Pieter Heyvaert, Pieter Colpaert,
Ruben Verborgh, Erik Mannens and Rik Van de Walle

Ghent University - iMinds - Multimedia Lab
First author: pheyvaer.heyvaert@ugent.be,
{firstname.lastname}@ugent.be

Abstract. Data Catalog Vocabulary (DCAT) is a W3C specification to describe datasets published on the Web. However, these catalogs are not easily discoverable based on a user's needs. In this paper, we introduce the Node.js module 'dcat-merger' which allows a user agent to download and semantically merge different DCAT feeds from the Web into one DCAT feed, which can be republished. Merging the input feeds is followed by enriching them. Besides determining the subjects of the datasets, using DBpedia Spotlight, two extensions were built: one categorizes the datasets according to a taxonomy, and the other adds spatial properties to the datasets. These extensions require the use of information available in DBpedia's SPARQL endpoint. However, public SPARQL endpoints often suffer from low availability, its Triple Pattern Fragments alternative is used. However, the need for DCAT Merger sparks the discussion for more high level functionality to improve a catalog's discoverability.

Keywords: data publishing, DCAT, Triple Pattern Fragments, Linked Open Data, Open Data, smart cities

1 Introduction

DCAT[1], short for Data Catalog Vocabulary, is a W3C specification for describing data catalogs, using Linked Data. It is a rather small vocabulary which has three main classes: *dcat:Catalog, dcat:Dataset* and *dcat:Distribution*. A *dcat:Catalog* is a class which can be used to describe the entire catalogue, e.g., who is the maintainer, when was it created, when was it last updated, what is the license of the metadata, and so on. The *dcat:Dataset* is a class to describe a dataset, a set of data facts which is published through one or more *dcat:Distributions*. A *dcat:Distribution* describes in its turn how data can be retrieved from the dataset it belongs to.

2 Problem Statement & Proposed Solution

The data catalogs are distributed over the Web. However, this distributed nature does not make it straightforward to reuse the catalogs directly, because discovering the catalogs

[1] *http://www.w3.org/TR/vocab-dcat/*

(and its datasets), based on a certain need, is difficult for the following reasons:

1. it is not possible to query multiple feeds simultaneously,
2. different protocols are used for offering different feeds, and
3. additional information (e.g., themes and spatial coverage of datasets) provided by the different catalogs is not always interoperable, i.e. different ontologies are used.

In this paper, we introduce DCAT Merger[2], build using Node.js[3]. It aggregates DCAT feeds in one feed to solve the first two problems. It comes with three enrichment methods: determining the subjects and geographical areas of the datasets, and categorizing the datasets using a theme taxonomy. This is done using Named Entity Recognition (NER) on the descriptions and keywords of the datasets. These methods solve the third problem, and improve the general discoverability of the catalog.

3 Architecture

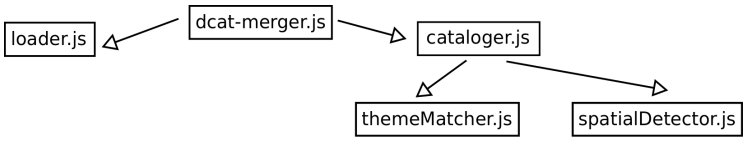

Fig. 1. The architecture of the DCAT Merger Node.js module.

DCAT Merger module consists out the following six files: *dcat-merger.js*, *loader.js*, *cataloger.js*, *themeMatcher.js* and *spatialDetector.js*. These files can be found in the folder */lib*. The entry point for the module is offered by *dcat-merger.js*, which uses *loader.js* and *cataloger.js*. The former is used to load the input feeds. Next, the latter merges them, which results in a single output feed. The *cataloger.js* makes use of *themeMatcher.js* and *spatialDetector.js* to enrich the output feed with theme and spatial information of the datasets.

4 Merging Feeds

First, we start by loading the different input feeds each in a separate triple store. Next, a triple store is created for the output feed, hence, containing the information about the new catalog. After adding the basic information about the catalog, we add the information about the datasets from each input feed's triple store. However, during this process, for each dataset the necessary adjustments to the triples are made, so that they connect to the newly created catalog. The triple store-functionality was provided by the Node.js module n3[4].

[2] *https://github.com/pheyvaer/dcat-merger*
[3] *https://nodejs.org/*
[4] *https://www.npmjs.com/package/n3*

5 Enriching Feeds

Besides only merging the different input feeds, we try to enrich them via three options: the *subjects* of the datasets, the *themes* of the datasets, and the *spatial coverage* of the datasets. We do this to improve the discoverability of (a group of) datasets in the catalog when keeping certain use cases and needs in mind.

5.1 Subjects

Based on the available keywords and descriptions, provided through *dcat:keyword* and *dcat:description*, we use NER to determine the (URI of the) subject(s) of each dataset. NER is facilitated by DBpedia Spotlight [3]. The request that is sent to the DBpedia spotlight server contains a string with a keyword or the description of a dataset. In return, a list of corresponding DBpedia resources is received, if any. This information is added to the resulting feed, hence, to its corresponding triple store.

5.2 Themes

The DCAT specification allows to denote the main themes (or categories) of a dataset using the property *dcat:theme*. Based on the subjects dissected in Section 5.1, we determine the themes of the datasets. To decide which themes are implicated by which subjects, we have created *themeMatcher.js*. It takes a subject as input and returns the corresponding theme, if any. All themes belong to the taxonomy defined at *http://ns.thedatatank.com/dcat/themes*. At the moment the mapping is manually defined. This information is added to the output feed's triple store. When using our module, the generation of themes is optional.

5.3 Spatial Coverage

If a subject, dissected in Section 5.1, refers to a geographical area, we also connect it to its dataset with the property *dcat:spatial*. To determine whether a subject represents a geographical area, we inspects its classes, i.e., check if the subject is an instance of the class *http://dbpedia.org/ontology/Place*. The relevant classes are configurable by the user. The functionality for detecting the spatial information is provided by *spatialDetector.js*. When using our module, the generation of the spatial coverage information is optional.

5.4 Triple Pattern Fragments

Both the *themeMatcher.js* and *spatialDetector.js* need additional information besides the (DBpedia) URI of a subject. That is, the classes that the subjects belong to. This information is available in DBpedia, which is accessible through a SPARQL endpoint[5]. However, the availability of such an endpoint is questionable [1]. That is why we opted

[5] *http://dbpedia.org/sparql*

to use its Triple Pattern Fragments (TPF) [4] alternative[6]. Using Node.js, there is a TPF client available through the module *ldf-client*[7].

6 Real-World Application: OTN

OpenTransportNet (OTN) is a project granted by the European Commission's CIP-ICT-PSP 2013-7 Call. By bringing together open geo-spatial data within City Data Hubs and enabling it to be viewed in new easy to understand ways, OTN enables new reuse of existing open datasets. In order to keep a relevant list of datasets, DCAT Merger was configured with various DCAT sources, and a new DCAT feed was generated for a certain city. The data was afterwards loaded in a virtuoso triple store, and published using The DataTank [2]. A demo of this can be viewed at *http://ewi.mmlab.be/otn/*.

7 Conclusion & Future Work

In practice, the use of DCAT Merger allows to create a single DCAT feed, which improves finding the catalogs that satisfy a user's needs. It is possible to enrich the feed with theme and spatial information, next to subject information. However, adding other custom extensions to DCAT Merger involves adding and changing the code in multiple places. To this extent, a plugin system should be developed to circumvent this. As a result, the functionality provided by *themeMatcher.js* and *spatialDetector.js* should be added as plugins. However, the need for DCAT Merger raises the questions if more (high level) functionality is required, e.g. on the server side, to solve the problems addressed by our module.

References

1. C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche. SPARQL web-querying infrastructure: Ready for action? In *The Semantic Web–ISWC 2013*, pages 277–293. Springer, 2013.
2. P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. Painless URI Dereferencing Using the DataTank. In V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, and A. Tordai, editors, *The Semantic Web: ESWC 2014 Satellite Events*, volume 8798 of *Lecture Notes in Computer Science*, pages 304–309. Springer International Publishing, 2014.
3. P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia Spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.
4. R. Verborgh, O. Hartig, B. De Meester, G. Haesendonck, L. De Vocht, M. Vander Sande, R. Cyganiak, P. Colpaert, E. Mannens, and R. Van de Walle. Querying datasets on the Web with high availability. In *The Semantic Web–ISWC 2014*, pages 180–196. Springer, 2014.

[6] Basic Triple Pattern Fragments server of DBPedia is available at *http://fragments.dbpedia.org*.
[7] *https://www.npmjs.com/package/ldf-client*