DataID: Towards Semantically Rich Metadata For Complex Datasets

Martin Brümmer
bruemmer@informatik.uni-leipzig.de

Ciro Baron
cbaron@informatik.uni-leipzig.de

Ivan Ermilov
iermilov@informatik.uni-leipzig.de

Markus Freudenberg
markus.freudenberg@gmail.com

Dimitris Kontokostas
kontokostas@informatik.uni-leipzig.de

Sebastian Hellmann
hellmann@informatik.uni-leipzig.de

AKSW/BIS, Universität Leipzig
PO Box 100920, 04109 Leipzig, Germany

ABSTRACT

The constantly growing amount of Linked Open Data (LOD) datasets constitutes the need for rich metadata descriptions, enabling users to discover, understand and process the available data. This metadata is often created, maintained and stored in diverse data repositories featuring disparate data models that are often unable to provide the metadata necessary to automatically process the datasets described. This paper proposes DataID, a best-practice for LOD dataset descriptions which utilize RDF files hosted together with the datasets, under the same domain. We are describing the data model, which is based on the widely used DCAT and VoID vocabularies, as well as supporting tools to create and publish DataIDs and use cases that show the benefits of providing semantically rich metadata for complex datasets. As a proof of concept, we generated a DataID for the DBpedia dataset, which we will present in the paper.

Keywords

metadata, documentation, dcat, void, provenance, dbpedia

1. INTRODUCTION

In 2011, the European Commission published its Open Data Strategy[1] defining the following six barriers for "open public data":

a) a lack of information that certain data actually exists and is available
b) a lack of clarity of which public authority holds the data
c) a lack of clarity about the terms of re-use
d) data which is made available only in formats that are difficult or expensive to use
e) complicated licensing procedures or prohibitive fees
f) exclusive re-use agreements with one commercial actor or re-use restricted to a government-owned company

While price, licensing and terms of re-use often are economical decisions made by the data publishers and therefore cannot be addressed in a research paper, the lack of information on the existence and availability of data as well as the clarity of its provenance are pressing issues the LOD community has to face.

Linked Open Data offers the unique chance of accessing vast amounts of machine-readable, semantically annotated data. However, access is still limited by additional knowledge required for data discovery. Data consumers have to know where datasets of interest are located, what kind of data they contain, where to access them in which formats, as well as the terms of reuse. To date, some parts of this important metadata can be found in various repositories,`datahub.io` being the most accepted one, although various domain-specific repositories[2] exist. Data models of these repositories vary and none of them offer enough granularity to sufficiently describe complex datasets in a semantically rich way. For example, `datahub.io` partially implements the Data Catalog Vocabulary (DCAT) W3C Recommendation[3], but only describes all resources associated with a dataset superficially, be it an ontology, a single example file, a diagram or a data dump as a distribution of the dataset. Most additional properties described are simple key-value pairs linked by `dcterms :relation` properties. This data model is semantically poor and prohibitive for most use cases wanting to automatically consume the data of a dataset. The DBpedia dataset, with its different versions and languages, multiple SPARQL endpoints and thousands of dump files with various content serves as one example of the complexity metadata models need to be able to express. We argue that the DCAT vocabulary as well as the established VoID vocabulary only provide a basic interoperability layer to discover data. In

[1] `http://europa.eu/rapid/press-release_MEMO-11-891_en.htm`

[2] for example MetaShare (`http://www.meta-net.eu/meta-share/index_html`) and Clarin (`http://clarin.eu/`)

[3] `http://www.w3.org/TR/vocab-dcat/`

SEM '14 September 04 - 05 2014, Leipzig, AA, Germany
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2927-9/14/09 ...$15.00.
http://dx.doi.org/10.1145/2660517.2660538

their current state, they still have to be expanded to fully describe datasets as complex as DBpedia (cf. (see Section 2), one of the core datasets of the LOD cloud.

Especially three important aspects are underspecified in these vocabularies: (1) PROVENANCE: a crucial aspect of data and needed to assess correctness and completeness of the data conversion, as well as the trustworthiness of the data source. (2) LICENSING: Machine-readable licensing information is as important, as it provides the possibility to automatically process and publish only data that explicitly allows these actions. (3) ACCESS: Finally, publishing and maintaining this kind of metadata together with the data itself serves as LOD-compatible documentation benefiting the potential user of the data as well as the creator by making it discoverable and crawlable.

In this paper, we will tackle these challenges by describing and implementing a dataset description best-practice based on widely accepted vocabularies for dataset metadata in RDF which we call DataID. We present our solution for an accessible, compatible and granular best-practice of dataset description, that improves on previous work and solves the issues presented here, including: (1) A vocabulary for dataset description based on DCAT, VoID, DCTerms, Prov-O and several extensions; (2) A tool to easily generate DataID files with a simple web form; (3) A configurable tool and web-service to upload the DataID to dataset repositories such as `datahub.io`; (4) The DBpedia 3.9 DataID as a proof-of-concept of how to implement the practice and completely describe the DBpedia dataset in all languages and variants, e.g. external links or wikidata extraction, comprising almost 20,000 different files for version 3.9.

2. RELATED WORK

In [6] the authors introduce a standardized interchange format for machine-readable representations of government data catalogues: DCAT vocabulary. At the moment of writing DCAT is available as a W3C recommendation[4]. Vocabulary terms for DCAT are inferred from the survey on seven data catalogs from Europe, US, New Zealand and Australia. The DCAT vocabulary includes the special class *Distribution* for the representation of the available materializations of a dataset (e.g. CSV file, an API or RSS feed). However, these distributions or parts thereof cannot be described further within DCAT. Applications which utilize the DCAT vocabulary (e.g. CKAN) in turn provide no standardized means for describing more complex datasets. This can be attributed to the fact that in 2010 at the time when DCAT was originated, the open data initiative was not mature enough and available machine readable formats were represented only by CSV, RDFa and feeds. Therefore, the problems for describing RDF datasets within data catalogs could not be foreseen.

The Vocabulary of Interlinked Datasets (VoID) [1] solves the problem of the discoverability of the datasets by providing metadata about them as a separate RDF datasets. VoID is widely accepted and used within the Semantic Web community, for instance in projects such as: OpenLink Virtuoso [5], LODStats [6], World Bank [7] and others. VoID can be used to express general metadata, access metadata, structural metadata and links between datasets. Tools to create VoID metadata are described in [2] where authors also presents techniques of reduction in order to create descriptions for Web-scale datasets. In the same paper the importance of VoID is well established but there is still a lack of important metadata which is not described, for example license and provenance. For simple datasets VoID performs well, which is supported by the fact of wide acceptance of the vocabulary. However, in a case of complex datasets VoID is not expressive enough. In particular, access metadata includes the *void:dataDump* property, which points to the data dumps of the particular dataset. However, this property should link directly to dump files as described in W3C Interest Group Note: Describing Linked Datasets with the VoID Vocabulary [8]. Thus additional semantic information about the data dumps and the structure of the dataset can not be expressed using VoID. Without such additional semantic information about the structure of an RDF dataset it has to be analysed by the human user before processing.

The Open Digital Rights Language[9] is an initiative of a W3C community group, aiming to develop an open standard for policy expressions. The ODRL version 2.0 core model defines 8 classes to define licensing policy in regard to the permissions it grants and the duties and constraints associated with these permissions as well as involved legal parties. Thus, an ODRL description allows to specify in a machine-readable way if the data can be edited, redistributed and re-purposed and which are the constraints.

Due to the complicated nature of converting legal texts into a number of logical axioms, existing ODRL resources should be used if possible. The ontology engineering group of the Universidad Politécnica de Madrid provides a number of these descriptions[10], in particular all current Creative Commons licenses, Open Data Commons as well as GPL, thus covering the most widely used Open Data licenses.

The Provenance Ontology (Prov-O)[11] is a W3C-recommended ontology for provenance description and widely adopted as a light-weight way to granularly express the source of data, its processing activities as well as involved actors. VoID and DCAT by themselves only use basic metadata to express provenance based on *DCTERMS*[12] properties like `dcterms:creator` or `dcterms:source`, thus not natively supporting further description of the involved persons needed to create the described dataset. There also is no support or incentive to describe source datasets and conversion activities, restricting reproducability and examination of conversion exhaustiveness. This lack is crucial from a scientific perspective, as it hinders replication of the data transformations common to LOD datasets.

The LODStats application described in [3] is a web appli-

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

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

[6] http://stats.lod2.eu/

[7] http://worldbank.270a.info/

[8] http://www.w3.org/TR/void/

[9] http://www.w3.org/community/odrl/

[10] http://purl.oclc.org/NET/rdflicense

[11] Prov-O: http://www.w3.org/ns/prov#

[12] DCMI Metadata Termshttp://purl.org/dc/terms/

cation for collection and exploration of Linked Open Data statistics. LODStats consists of two parts: *the core* collects statistics, such as ontologies used, number of triples and links between datasets about datasets of the LOD cloud and publishes it on the LODStats web portal, a *front-end* for exploration of dataset statistics. Statistics are published both in human-readable and machine-readable formats, thus allowing consumption of the data through the web front-end by the users as well as through an API by services and applications. The main deficiency of the LODStats application is that it relies on the DCAT data model, which is not descriptive enough for complex datasets. More insights on LODStats and DataIDs as well as the future work on the LODStats in relation with DataID data model are provided in Section 8.

3. DATAID DATA MODEL

The DataID data model provides a uniform way to describe general metadata of datasets. In particular, the data model is compliant with `datahub.io` but improves upon it by including richer semantics for properties relevant to LOD datasets.

Figure 1 shows the overall structure of DataID. As discussed in Section 2, there are different vocabularies which can be used to describe a dataset. *VoID* defines a central class `void:Dataset`, properties particular to RDF datasets like `void:triples` and `void:sparqlEndpoint` as well as criteria on how to use other vocabularies, one of them being *DCTERMS*. The DataID data model uses VoID extensively and includes the RDF specific properties. It also includes the property `void:subset` to introduce descriptions of parts of datasets. This is especially important to describe monolingual subsets of multilingual datasets. The class `void:Linkset` was adopted as well, enabling the description of content and number of links between different datasets. Using linksets, visualizations that show connections between datasets, like the ubiquitous LOD cloud diagram[13] could be easily realized without having to directly access and process the data.

DCAT also defines a dataset class, `dcat:Dataset`. Like VoID, it also uses DCTERMS properties for general metadata, thus enabling us to merge the dataset concept of both vocabularies into `datid:Dataset`. Alongside minor additions, like `dcat:contactPoint` or `dcat:keyword`, DCAT defines the class `dcat:Distribution` for further description of directly accessible serializations of the data itself. This concept is crucial to be able to automatically retrieve and use the data described in the DataID, simplifying, for example, data analysis. To account for SPARQL-endpoints as special type of distributions, the *SPARQL Service Description* (SD)[14] vocabulary was used to differentiate it from file based distributions. Properties used include `sd:endpoint` to link to the endpoint's URL, `sd:resultFormat` to annotate the format of the results, as well as the class `sd:Service` as a type of distribution. Because describing the name of the graph that contains the dataset's data in SD would imply creating a `sd:Graph` resource and thus create too much overhead, `datid:graphName` was defined. It contains the name of the relevant graph as a literal string.

A critical gap in both vocabularies is granular provenance information, which is important because most Linked Data is derived from other datasets, either by conversion or aggregation. In order to make scientific use of Linked Data and to also properly attribute the original data creators, it is essential for a researcher to have as much provenance information as possible on the dataset. Commercial users need this information as well, if they want to include LOD in their product and need to address legal issues with the owners of the data. Neither VoID nor DCAT by themselves are able to express provenance information beyond single persons or related sources of the data.

The Provenance Ontology (Prov-O) with its classes `Entity`, `Agent` and `Activity` including their respective properties is used in DataID to capture a complete, fine-grained provenance chain. Entities serve as a super-class of datasets and are either other datasets, or, in general, all kinds of sources of which datasets are derived, like books or other primary sources. Agents are all kinds of persons involved with dataset creation or maintenance. Activities are mediators in the derivation, integration or aggregation of datasets from other datasets and sources, denoting in their description what changes were made in the process.

Besides the integration of existing vocabularies, DataID adds some additional properties to allow for convenient querying of metadata. For example, `datid:licenseName` contains the name of the license linked by the `dcterms:license` property, so the name of the license can be easily retrieved without querying another resource on a different server. Versioning was also implemented with DataID properties. A dataset can link to different versions by the `datid:version` property. Information about the dataset's version is given as a literal by the `datid:versionInfo` property. To easily be able to query the latest version of the dataset, the `datid:latestVersion` property should link to the respective resource. Finally, `datid:containsLinks` is a subproperty of `void:subset` allowing to query for linksets without having to specify the `rdf:type` of the linked resource. Beyond the integrated data model, DataID features *OWL* axioms to define mandatory and optional properties in dataset description through cardinality restrictions.

Further integration with LODStats and SparqlES will make it possible to create a module that will automatically generate statistical data from datasets (like triple count, SPARQL service uptime, Ontology Usage and links to other datasets).

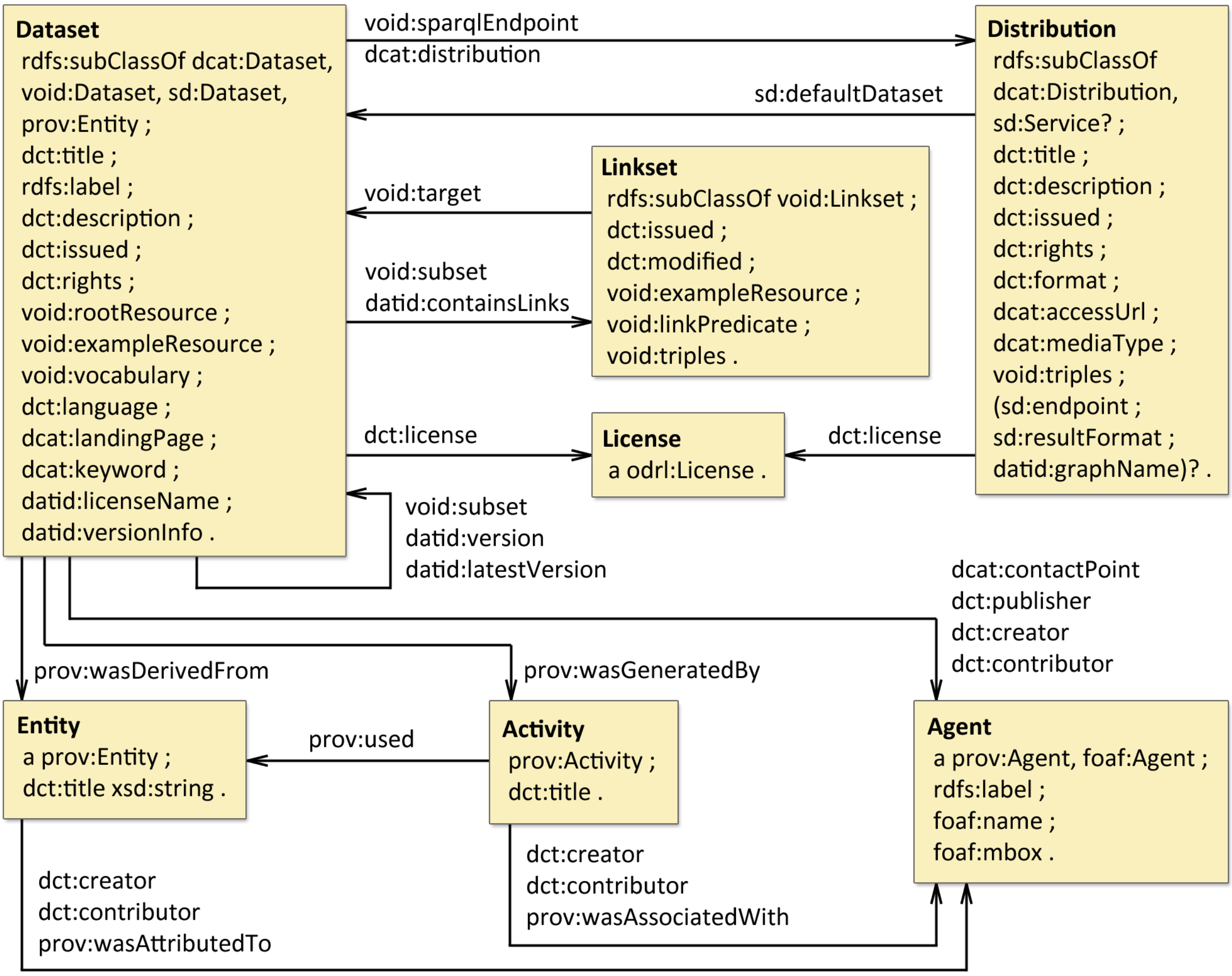
4. IN USE

In practice, metadata is only as good as its availability and discoverability. Besides the well-defined data model, the deployment of DataID descriptions has to be determined and implemented by its users. To ease adoption and be as inclusive as possible, we are following the `robots.txt` convention. This widely implemented de-facto standard[15] consists of a file `robots.txt` in the top-level directory of a web server that contains a number of constraints regarding which URLs of the web site are forbidden from being visited by programs automatically crawling it. An advantage of the `robots.txt` is its ease of deployment. The simple file can just be put

[13] http://lod-cloud.net/

[14] http://www.w3.org/TR/sparql11-service-description/

[15] http://www.robotstxt.org



@prefix dct: <http://purl.org/dc/terms/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix sd: <http://www.w3.org/ns/sparql-service-description#> .
@prefix void: <http://rdfs.org/ns/void#> .
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix datid: <http://dataid.dbpedia.org/ns#> .

Figure 1: DataID vocabulary diagram

into the relevant directory; no scripting, redirecting or other installation is necessary. Similarly, we propose DataID files to be put in the top-level directory of the web server. The format must be Turtle, which, in our opinion, is the RDF serialization featuring the best compromise between readability and file size. The file should simply be called `dataid.ttl`. In example, next to the `http://dbpedia.org/robots.txt` that explicitly excludes certain directories from being crawled, the `http://dbpedia.org/dataid.ttl` explicitly states where DBpedia datasets can be found as well as their content and relevant metadata. This best-practice allows publishers to aggregate descriptions of all hosted datasets in one place and also enables users to easily discover and access these datasets. It facilitates easy aggregation of an institution's or project's datasets and massively cuts down on the time spent on navigating and searching for relevant datasets on diverse web sites.

Furthermore, we encourage using content negotiation via 303 redirects to forward to the DataID location when the root domain itself is queried for RDF data, but understand that it is hard to implement[16] for some users and therefore opted to simply deploying the DataID description file in a directory on the server as a default. Dataset URIs in the DataID can be realized by adding a hash to the DataID files' URI, as well as using URNs. The following example will demonstrate the practice and show the DBpedia as in-use example.

4.1 DBpedia DataID Example

In order to test that the DataID is powerful enough to express the structure and metadata of complex datasets, we created a DataID for the DBpedia 3.9 dataset[17]. The DBpedia is an aggregation of multiple datasets, one for each language extracted. At the moment, 119 language editions of the DBpedia exist. 15 languages have their own DBpedia chapter [4], providing own maintaining personal, own hosting solutions and subdomains, published LOD as well as SPARQL endpoints. One of these chapters represents the *DBpedia core*, as it is materialized under the `http://dbpedia.org` domain. The core version is unique in aggregating multiple languages as well as secondary datasets, for example those containing links to other datasets. However, the data the DBpedia core aggregates is not identical with the data the other chapters distribute. Chapters have shorter release cycles than the DBpedia core and are versioned differently, because they maintain a smaller amount of data, namely only one language. All core language versions can be accessed by downloading around 170 dump files each, accessible in 4 different formats[18] each under `http://downloads.dbpedia.org`. These files contain topical subsets of the main dataset, for example only the triples featuring the `rdf:type` or `rdfs:label` of the resources.

The DBpedia core 3.9 DataID contains 121 `datid:Dataset` resources, one for each language as well as a link dataset and the Wikidata extraction dataset, coinciding with the folders at `http://downloads.dbpedia.org/3.9/`. An example resource of the DBpedia 3.9 DataID can be seen in Listing 1. The example reveals that there is some redudancy in titles and descriptions, which is rooted in missing metadata that are filled by automatic generation of these describing strings. Other properties are deliberately redundant, such as using both `dcat:accessURL` and `dcat:downloadURL` to express the location of a distribution. We made this decision to increase compatibility and interoperability with tools automatically processing the DataID that are trying to access relevant resources.

At the moment, chapters other than the core are only described by a dataset resource and a SPARQL endpoint distribution, because there is no procedure to aggregate the metadata over all different chapters. In the future, the chapters will maintain their only DataIDs, alleviating this issue.

As for the other languages, because there is no single file that aggregates the complete content of a DBpedia language edition, dataset resources contain a number of `void:subset`s describing the sub datasets that are composed of the individual dump files' contents. These sub dataset resources in turn link to their distribution's resources, that describe the individual dump files, one for each format available. Because the same is true for the DBpedia core, the datasets loaded into the public SPARQL endpoint at `http://dbpedia.org/sparql` can easily be obtained by downloading the relevant dump files, providing an easy way to mirror the DBpedia. The total number of languages and distribution files in the DBpedia makes for a very large DataID, containing 4591 datasets in 119 languages, materialized in 16244 distributions.

Compiling the data needed for the DataID was not an easy task. Due to the high number of dump files, the server directories had to be crawled to automate the creation of distribution resources and associated datasets. Other metadata is aggregated on a number of Wiki pages, for example `http://wiki.dbpedia.org/DatasetsLoaded39` that states which datasets are part of the DBpedia core 3.9. There also is a Github repository[19] that documents the provenance of external links. Other metadata is scattered over a number of scientific publications as well as e-mail exchanges. For older DBpedia versions, necessary metadata got either lost or was never properly recorded, so that a complete DataID for all DBpedia versions back to 1.0 can only be created with a lot of effort.

Provenance information on the datasets still is sparse and simply does not completely exist before DBpedia version 3.7, meaning the specific version of the Wikipedia dumps used to convert the data into RDF is not known any more. The version of the DBpedia Extraction Framework used to compile these datasets also was not documented. Even in the case where it is known which version of Wikipedia dump was used, they are not available any more at their source location[20]. It is therefore impossible to assess completeness and correctness of the DBpedia conversion because neither the source data nor the software used are available for replication. This may not be important for the DBpedia as a compilation of encyclopedic knowledge, but highlights the

[16] content negotiation imposes a higher know-how barrier as well as extended access rights

[17] `http://dbpedia.org/dataid.ttl`

[18] Turtle, NTriples, N-Quads and TQL

[19] `http://github.com/dbpedia/dbpedia-links`

[20] `http://dumps.wikimedia.org/`

overall importance of keeping provenance information and potentially mirroring source data for scientific purposes.

```
@prefix : <http://dbpedia.org/dataid.ttl#>

:DBpedia_en_3.9
 a dcat:Dataset, void:Dataset, sd:Dataset, prov
     :Entity ;
 dct:title "DBpedia English" ;
 rdfs:label "DBpedia" ;
 dct:description "DBpedia is a ..." ;
 dcat:keyword "lod" , "rdf" , "wikipedia" ;
 dct:issued "17-Sep-2013"^^xsd:date ;
 dct:publisher <http://wiki.dbpedia.org/
     Association> ;
 void:exampleResource <http://dbpedia.org/data/
     Leipzig.rdf> ;
 dct:language "en" ;
 dcat:distribution <http://dbpedia.org/sparql>
     ;
 void:sparqlEndpoint <http://dbpedia.org/sparql
     > ;
 dcat:landingPage <http://dbpedia.org/> ;
 prov:wasDerivedFrom <http://dumps.wikimedia.
     org/enwiki/20130403> ;
 prov:wasGeneratedBy :DBpedia_Extraction ;
 dct:license <http://creativecommons.org/
     licenses/by-sa/4.0/rdf> ;
 dct:rights "DBpedia 3.9 is licensed under the
     terms ..." ;
 datid:ontologyLocation <http://downloads.
     dbpedia.org/3.9/dbpedia_3.9.owl> ;
 datid:licenseName "Creative Commons
     Attribution-ShareAlike 4.0" ;
 datid:versionInfo "3.9" ;
 dcat:contactPoint <http://wiki.dbpedia.org/
     Association> ;
 datid:latestVersion :DBpedia_en_3.9 ;
 void:subset <http://dbpedia.org/dataid.ttl#
     DBpedia_en_3.9_article_categories_en> .

:DBpedia_en_3.9_article_categories_en
 a dcat:Dataset, void:Dataset, sd:Dataset, prov
     :Entity ;
 dct:title "DBpedia English article categories"
      ;
 rdfs:label "DBpedia" ;
 dct:description "DBpedia English article
     categories" ;
 dct:issued "17-Sep-2013"^^xsd:date ;
 dct:modified "10-Aug-2013"^^xsd:date ;
 dct:publisher <http://wiki.dbpedia.org/
     Association> ;
 dct:language "en" ;
 dcat:landingPage <http://dbpedia.org/> ;
 prov:wasDerivedFrom <http://dumps.wikimedia.
     org/enwiki/20130403> ;
 prov:wasGeneratedBy :DBpedia_Extraction ;
 dct:license <http://creativecommons.org/
     licenses/by-sa/4.0/rdf> ;
 dct:rights "DBpedia 3.9 is licensed under the
     terms ..." ;
 datid:ontologyLocation <http://downloads.
     dbpedia.org/3.9/dbpedia_3.9.owl> ;
 datid:licenseName "Creative Commons
     Attribution-ShareAlike 4.0" ;
 datid:versionInfo "3.9" ;
 dcat:contactPoint <http://wiki.dbpedia.org/
     Association> ;
 datid:latestVersion :DBpedia_en_3.9 ;
 dcat:distribution <http://downloads.dbpedia.
     org/3.9/en/article_categories_en.nq.bz2>,
     <http://downloads.dbpedia.org/3.9/en/
     article_categories_en.nt.bz2> .

<http://downloads.dbpedia.org/3.9/en/
    article_categories_en.nq.bz2>
  a dcat:Distribution ;
  dct:title "article categories en" ;
  dct:description "DBpedia dumpfile:
      article_categories_en.nq.bz2" ;
  dct:issued "17-Sep-2013"^^xsd:date ;
  dct:modified "10-Aug-2013"^^xsd:date ;
  dct:license <http://creativecommons.org/
      licenses/by-sa/4.0/rdf> ;
  dcat:accessURL <http://downloads.dbpedia.org
      /3.9/en/article_categories_en.nq.bz2> ;
  dcat:downloadURL <http://downloads.dbpedia.
      org/3.9/en/article_categories_en.nq.bz2>
      ;
  dcat:byteSize "270000000"^^xsd:decimal ;
  dcat:mediaType "application/x-bzip" ;
  dct:format  "nq" .
```

Listing 1: Example DataID excerpt for the DBpedia 3.9 core dataset

5. DATAID GENERATOR

The creation of RDF files with large amounts of metadata can be a difficult task for users that have no technical knowledge or experience in Semantic Web technologies. In order to help people from other domains create DataID files in a convenient way, a DataID generator[21] was developed. The generator consists of a web application with a set of input boxes that, after having been filled, will provide a Turtle DataID file. Based on stable frameworks such as AngularJS[22] and Bootstrap[23], and using PHP EasyRDF[24], the DataID generator can be used in most browsers. Hence it's possible to use the advantages of DataID in a visual way, which makes it easier to understand the basic structure of a DataID file. Users are also not forced to manually edit Turtle files, which might not be a problem to advanced users, but makes metadata generation much more accessible for less experienced users.

Using the generator also implies that several structural or syntax errors can be avoided, which can easily happen when an RDF file is created manually. The user only needs knowledge of the metadata values and this is sufficient to create a valid DataID file. Besides avoiding structural errors, using the generator offers some facilities that are not available when creating files manually. For instance, it is possible to fill the number of triples (`void:triples`) of a distribution automatically fetching statistic data from LODStats (see Section 8). Other advantages include the possibility of automatically uploading DataID file to Datahub, using the Datahub mapper which is described in Section 7. There are also implicit advantages like preventing users from adding invalid URIs, as they are validated on the fly, avoiding inconsistencies choosing predefined values of properties (e.g. `odrl:License`) and preventing users to type invalid literal values for typed fields such as dates.

The generator interface allows a non-expert user to visualize the basic structure of DataID. Once the application is opened, an initial dataset is automatically created, allowing the user to complete required metadata. From this initial dataset, the user can create multiple subsets comprising a specific domain of the dataset in question. A subset may contain multiple distributions, which are dumps of datasets and contains properties such as the amount of triples and file extension. Figure 2 shows an activity diagram that illustrates the basic functionality of the generator. Buttons allow to

[21] http://dataid.dbpedia.org/

[22] https://angularjs.org

[23] http://getbootstrap.com/

[24] http://www.easyrdf.org/

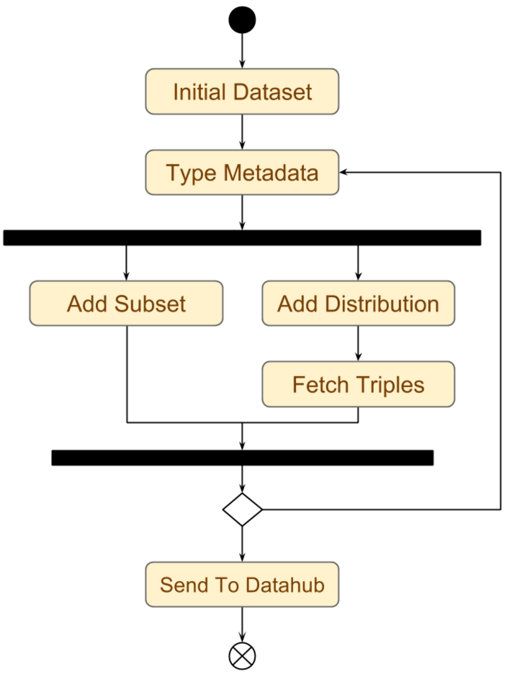


Figure 2: Activity Diagram for DataID Generator

fetch the number of triples from LODStats when creating a distribution, or to automatically send the DataID file to Datahub.

The DataID generator can be used to create small to medium dataset files with a small number of distributions and subsets. The main difficulty of creating file for large datasets is that too much manual effort is required. The generator is therefore only intended for manageable dataset sizes for which it can be a very helpful tool. In contrast, datasets like the DBpedia have in the order of thousands of different distributions, which makes it impracticable to manually create each of these distribution even using the advantages of the generator.

In these cases the generator can be used as a starting point where more metadata can be included from other sources, like crawled wiki pages or server directories. So the RDF file generated by the generator can be used as the basis for creating DataIDs for larger dataset. An alternative solution might be to create DataID files through queries that crawl datasets in repositories and other sources.

6. DATAID VALIDATOR

In addition to the DataID generator, we provide a DataID validation service[25]. Validation is an important factor for a new standard adoption and eases the generation of valid DataID descriptions. The *DataID Validator* implementation is based on RDFUnit[26]. RDFUnit follows a test-driven approach [5] for evaluation of Linked Data that is inspired from test-driven software development. For every axiom in the DataID ontology, a SPARQL test case is generated and ensures that the axiom is not violated, following a closed world assumption and a weaker form of the unique names assumption. Moreover, additional manual test cases are defined performing quality checks that cannot or should not be expressed in RDFS and OWL. Such manual test cases can report warnings or improvement suggestions. For example, every dataset should have at least one subset that is a void:Linkset to properly denote it's links to other datasets, making it part of the LOD cloud, furthering exploration and making it 5 star data[27]. However, there may be cases where the data cannot feasibly be linked to other datasets and this fact should not preclude the dataset from being described with a valid DataID. Thus, a manual test that checks for the existence of a void:Linkset and issues a warning in case none exists, is executed as part of the validation.

7. DATAHUB.IO INTEGRATION

In the recent past datahub.io has emerged as a linchpin of the Open Knowledge Foundation[28]. Providing a web interface for publishing and searching any kind of data-sources, datahub.io aggregates information about data. DataID itself does not natively provide a repository with search functionality, which was considered imperative in making DataID information accessible to a wider audience. datahub.io, as a widely used platform, was chosen to satisfy the demands for an open, easy to access and well-known repository environment.

Based on the CKAN data management system[29], also developed by the Open Knowledge Foundation, datahub.io provides a simple access to features like search and faceting of data-sources. CKAN makes the whole range of functions related to the management of datasets accessible via a REST interface. Thus, compatibility of the DataID approach with datahub.io is achieved by a simple REST client to publish DataID metadata on datahub.io, that uses the create and update functions for datasets. To provide the information needed to declare a datahub.io dataset, properties of the DataID ontology have to be mapped to match properties used by datahub.io and CKAN. A property mapper[30] was implemented to dynamically map any property by using a mapping configuration file. This configuration file uses JSON-LD [31] to provide structural, contextual and easy-to-read mappings between DataID and datahub.io properties. Listing 2 shows an example mapping for the property "title" of the class "dataset". Mappings consist of a @graph object, containing the repository specific mappings. These mappings in turn contain objects for all mappable classes of repository, in the example Datahub's dataset object, which then contain JSON-LD objects for all mappable properties of the object, like title. The @id node identifier then defines the DataID property that is mapped to the respective repository property.

```
{"@graph":
  {"dataHubMapping":{
    "dataset":{
      "title": {
         "@id": "dc:title",
         "@type": "xsd:string",
```

[25] http://dataid.dbpedia.org/validate
[26] http://rdfunit.aksw.org
[27] http://www.w3.org/DesignIssues/LinkedData.html
[28] https://okfn.org
[29] http://ckan.org/
[30] Available at https://github.com/dbpedia/dataid
[31] http://json-ld.org

```
            "comment": "A name given to the
                dataset.",
            "addedBy": "Chile",
            "issued": "2014-04-16"
        }
    }
}}}
```

Listing 2: Example mapping configuration

This mapping can be extended to work for other repositories, thereby allowing to distribute updated metadata of a dataset quick and easy to all relevant repositories, minimizing maintenance. Furthermore, a REST service was added to use the described functionality directly as a web service, without deploying own software. Using this service, a new `datahub .io` dataset can be created by adding a DataID as data to an HTTP-POST with added parameters for the name of the dataset, the `datahub.io` organization and the user's API key. The service is also integrated into the DataID generator, further facilitating ease of use.

8. CONCLUSIONS AND FUTURE WORK

The creation and dissemination of DataIDs will further advance discoverability of the RDF datasets. DataID can help to improve metadata for new datasets, as it can be used to aggregate and validate important metadata right at the dataset creation. We are aware that the strict requirements of DataID and its validator impose additional burden on the data providers. On the other hand, these requirements will help user to analyse completeness of their metadata and guide them to improve it before the information is lost. Such profound descriptions will also help potential users (e.g. developers, analysts and data journalists) to select the required segments of the dataset, such as chapters of DBpedia datasets or particular serializations of specific parts of the dataset. Thus, it enables sophisticated filtering over the dataset. We plan to exploit this feature of the DataID to extend the applicability of the LODStats[32] portal. The main problem is, that at the time of writing statistics for the datasets on the LODStats portal are partial. In particular, for each dataset the LODStats portal contains statistics based only on one data dump or SPARQL endpoint. Therefore, in cases of complex datasets that are sliced in several parts only partial statistics are available. To explain why only one data dump is evaluated for the given dataset we briefly describe an example of a complex dataset. B3Kat[33] dataset is sliced in six parts. Each part is represented on the `datahub.io` as a CKAN resource and has the same semantic meaning as its sibling resources: SPARQL endpoint, B3Kat download page, VoID description and example resource. Although a clear distinction between B3Kat download page and dataset slices exists, it is not explicitly defined and can only be inferred from the format of the resource. But the format of the resource is relatively arbitrary. Therefore, without a rich metadata description such as DataID it is not feasible to automatically determine all parts of the dataset which should be processed for a particular application like LODStats. On the other hand in the datasets described by DataIDs the relevant data dumps can be automatically identified and marked for the further processing with LODStats. In case of a versioned dataset, it is also possible to recognize the latest version of the dataset (i.e. with `datid:latestVersion` property) and automatically fetch new statistics, should it have changed.

Concerning the DataID generator, further improvements will be made. An important feature will be not only being able to upload DataID files to `datahub.io`, but also importing DataIDs. This way, the generator can be used as an editor, further easing metadata maintenance.

Visualization is another interesting aspect of DataID. After sufficient adoption, the `void:Linkset` resources of the datasets could be automatically crawled, automatically accessing DataIDs of linked datasets to traverse the graph of interlinked LOD datasets. Using this technique, a diagram of the LOD cloud could be created automatically, reflecting existing, working, published and properly described data, instead of being restricted to datasets that are published on `datahub.io`, like in the case of `http://lod-cloud.net/`.

We believe that working with Linked Open Data provides invaluable advantages to its users and producers. But these advantages are only tangible to users that can find and understand the data and producers that are able to properly manage and distribute their data. Using the DataID can help both sides achieve their goals and thus make Linked Open Data a more accessible way of publishing data.

Acknowledgements. We would like to thank Norman Heino for the fruitful discussion and the comment: ”Turtle syntax even looks a bit like the robots.txt”. This work was supported by grants from the EU's 7th Framework Programme provided for the projects LIDER (GA no. 610782), LOD2 (GA no. 257943).

9. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets. In *LDOW*, 2009.

[2] C. Böhm, J. Lorey, and F. Naumann. Creating void descriptions for web-scale data. *Web Semant.*, 9(3):339–345, Sept. 2011.

[3] I. Ermilov, M. Martin, J. Lehmann, and S. Auer. Linked open data statistics: Collection and exploitation. In *Knowledge Engineering and the Semantic Web*, pages 242–249. Springer, 2013.

[4] D. Kontokostas, C. Bratsas, S. Auer, S. Hellmann, I. Antoniou, and G. Metakides. Internationalization of linked data: The case of the greek dbpedia edition. *Web Semantics: Science, Services and Agents on the World Wide Web*, 15(0):51 – 61, 2012.

[5] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 747–758, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[6] F. Maali, R. Cyganiak, and V. Peristeras. Enabling interoperability of government data catalogues. In *Electronic Government*, pages 339–350. Springer, 2010.

[32] http://stats.lod2.eu/

[33] http://datahub.io/dataset/b3kat