**Investigation  
of linked open data technologies  
for purposes of publishing georeferenced statistical data**

Mirosław Migacz, Statistics Poland, [m.migacz@stat.gov.pl](mailto:m.migacz@stat.gov.pl)

**Abstract**

*Polish official statistics possesses a vast amount of statistical data dispersed among different databases and disseminated using various publication methods. While there is a significant increase in openness of the data, there is still a lot of work to be done in terms of integrating different data sources. That is why the Central Statistical Office of Poland decided to look into the linked open data technology. An ongoing project has resulted in an inventory of databases and data sources currently published by official statistics and is investigating linked data technologies in order to prepare a “cook-book” for a linked open data implementation.*

**Keywords:** linked open data, GIS, RDF, dissemination

**1. The linked open data pilot project**

In January 2018 Statistics Poland concluded the “Development of guidelines for publishing statistical data as linked open data” project. The aim of the project was to perform a thorough inventory of data sources and investigate technologies, which could be used to publish georeferenced statistics as linked open data.

Data samples from statistical databases and geospatial datasets have been selected for transformation to linked open data RDF triples and a dataset catalogue has been set up and encoded in RDF. A pilot triple store has been established with a SPARQL endpoint – a query interface. Aside from the pilot’s results being machine readable, all data created in the pilot was also internally published as human readable webpages using linked open data frontend software.

**2. Data scope**

*2.1. Statistical data*

A test scope of data from three major Statistics Poland’s databases (Local Data Bank, STRATEG, and Demography Database) has been selected for the purpose of the project. The data selected as a representative sample was the population by sex and age groups. Values for 348 variables were picked from the Local Data Bank, in the case of the Demography Database the number was 339, and in the case of the STRATEG database – 48. The selected data was used to designed age ontology for the aforementioned databases.

Samples of population by sex and age groups in 2016 for Poland and voivodships (the highest level of administrative division) were selected from the three above mentioned databases for conversion of statistical data to linked open data.

*2.2. Spatial data*

Statistical data chosen for publishing as linked open data contain a reference to territorial division identifiers. To make this data more usable, geometries of the territorial division units should also be published as linked open data. For purposes of this pilot, the geometries of Poland and voivodships have been selected for transformation.

*2.3. Data source catalogue*

The inventory of data sources carried out as one of the first stages of the project resulted in a structured catalogue of data sources described with metadata. Creating a dataset catalogue using linked open data was considered a valuable exercise for the pilot implementation.

**3. Designing ontologies**

The most important stage of designing a linked open data implementation is designing ontologies for published data sources. While it is at times necessary to design a whole new ontology for a data source, it is a good practice to re-use existing ontologies and vocabularies which have already been published. Publishing new ontologies which are similar to the ones already published should be avoided.

A thorough research of basic vocabularies as well as existing linked statistical data implementations is essential for designing useful ontologies. The core vocabularies are mostly stable and well described but during the course of the project it was very hard to find an implementation of statistical linked open data, which could be considered a reference. The biggest problem with existing implementations is that most of them have been published few years ago and have not been updated since. Not establishing repeatable processes in the organization to regularly revise and update the datasets results in most of the published data seeming abandoned, losing their value over time. Some of the implementations are internally inconsistent, e.g. it is apparent that different software tools have been used to publish different datasets and the resulting RDF files are not fully compliant with each other. Linking to other data sources is a vital part of linked open data. Sadly not all resources are being maintained and older implementations tend to have links to resources which are not available anymore.

Nevertheless every implementation is a valuable source of information on how to model statistical data using linked open data vocabularies.

In terms of statistical data the pilot focused on publishing demographic data on population by sex and age groups. Age groups used in publishing data on demography are usually country specific, that is why it has been decided to create a new ontology for the age classification. For the sex dimension an existing SDMX codelist was sufficient.

In terms of spatial data the pilot focused on creating an ontology for the Coding System for Territorial and Statistical Units (KTS), which comprises territorial units used for dissemination of statistical data in Poland. The Open Geospatial Consortium (OGC) GeoSPARQL[[1]](#footnote-1) standard was used to model relationships between classes and to encode geometries.

For cataloguing data sources identified in this project and datasets of statistical and spatial data, the DCAT Application Profile[[2]](#footnote-2) for data portals in Europe has been used. All datasets have been described with metadata, re-using existing vocabularies where possible, e.g. EuroVoc for thematic categories, EU Publication Office Continent and Country lists or Internet Media Type (MIME) vocabulary.

**4. Data transformation and software components**

*4.1. Encoding RDF metadata*

After designing ontologies for statistical classifications and spatial statistical units, these ontologies needed to be transformed into RDF metadata. First attempts were performed using Ontop software. An MS SQL Server 2008 database has been created and the designed ontologies were loaded to the database as tables. Then an attempt was made to define mappings between table fields and desired ontology classes (e.g. using SKOS, Dublin Core, RDF Schema vocabularies). The first results were not satisfactory – defining mappings to acquire a correct class and property references proved to be a time consuming process and some mappings could not be defined to match the designed ontologies. Also the RDF-XML files produced by Ontop were not generated as it was foreseen by the ontology designers. It was decided to try out other technologies before making further attempts to configure Ontop.

Python RDFlib package was chosen as an alternative tool for encoding RDF metadata. Main features of the package:

* tools to create RDF triples and store them in graphs,
* common namespaces such as RDF, RDFS or SKOS are already defined, other namespaces can be defined and bound to desired prefixes,
* parsing for RDF-XML, N3, NTriples, N-Quads, Turtle, TriX, RDFa and Microdata formats is possible, which allows transforming existing RDF metadata files to different formats, including RDFlib triples which show the exact syntax of the triple (subject, predicate, object). Parsing was especially useful for transforming files found on the Internet to learn how to construct triples and how they transform between different formats (e.g. RDF-XML and Turtle),
* serializing for all above mentioned formats is possible, so output files can be written in several formats.

Contrary to Ontop, RDFlib did not require a relation database input, so instead of a MS SQL Server database tables, simple tabulation separated text files were used. It is worth noting that if input files contain diacritics (e.g. Polish language specific letters), encoding of the input text file needs to be set / transformed into UTF-8 to ensure correct processing by RDFlib. Encoding can be checked / set in the open source Notepad++ software, which is also useful for viewing and analysis of XML files (e.g. RDF-XML).

The single most important advantage of the RDFlib Python package is the possibility to construct triples in any desired way (without the need for arduous configuration of mappings). Other advantages include the possibility to easily create output files in different formats and to modify Python scripts while working with the trial-and-error method or to supplement them later with links to other data sources or new vocabularies. All resulting RDF metadata in the pilot project was created using Python scripts.

*4.2. Triple store and SPARQL endpoint*

Apache Jena Fuseki software was used as a SPARQL server. Fuseki functions a both: a triple store and a SPARQL endpoint. All RDF graphs created within this project have been serialized and exported in both: RDF-XML and Turtle (TTL) format. Fuseki supports upload of both file types. All data was loaded into the triple store using the RDF-XML files. This project resulted in creating **71717 triples**. All triples have been loaded as a single Fuseki dataset to allow cross-querying and cross-browsing data created initially in separate files.

*4.3. Linked open data frontend*

Linked open data loaded into a SPARQL server allows data querying. Query results, which are URIs are provided as links. To make these links resolvable, a linked open data frontend needs to be set up. For this purpose Pubby software was used. Pubby creates webpages for each local URI defined in the datasets uploaded to the Apache Jena Fuseki SPARQL server. Each URI created within the pilot project could be viewed with all its associated properties as a webpage, which allows browsing through all published linked data entities.

**5. Conclusions**

The pilot project provided valuable knowledge on linked open data technologies and vocabularies. Several conclusions emerged:

* **No reference implementation for statistical linked open data.** There is no implementation of statistical data as linked open data that can be considered a fully correct, reference implementation. There are several existing implementations but most are plagued with some of the following issues:
  + lack of integrity between RDF metadata sets published by one authority – probably due to different software or programming components used,
  + links to non-existing entities – some implementations link to ontologies that have been published some time ago but are not online anymore,
  + lack of maintenance – most implementations are being developed and published but not later maintained (they contain data only for a certain year and URI persistence of data being linked to is not checked regularly).
* **Lack of pan-European guidelines for statistical linked open data.** Currently there is no guidelines for providing statistical data as linked open data (e.g. which vocabularies or software components to use), however several initiatives run under Eurostat’s DIGICOM project might result in a linked open data “cook book” in the near future.
* **Software / programming components not being developed anymore.** Some of the tools tested within this project (e.g. Ontop or Pubby) are not developed anymore, so their implementations might become unstable in time. Python RDFlib package seems sustainable at this point (triples are produced based on encoding subject-predicate-object statements which are then serialized in stable formats like RDF, TTL), but it is also not developed anymore.
* **Not much data to link to.** Linked open data makes most sense if it is connected with as much other data sources as possible. This project utilized several existing vocabularies and already published datasets (like reference years from the UK Government, EuroVoc or EU Publication Office’s country / continent lists) but a reference statistical linked open data implementation would be a much more desired resource to link to.
* **Semantic harmonization of statistical classifications.** This is not only a pan-European issue. It may also exist on the country level, if several datasets have different meanings for the same classification elements (e.g. difference in interpretation of age groups: 0-5 can be “0 to 5” or “0 to less than five”). Harmonization is always a difficult and complex issue, hopefully some conclusions in this matter will emerge from pan-European activities in statistical linked open data.
* **Methodology for publishing spatial data as linked open data.** This topic has two aspects: technological and temporal. In terms of technology, GeoSPARQL seems to be the correct way to publish spatial data as linked open data. The temporal aspect is much more complicated. In this project it has been decided to support publishing separate statistical unit geometries for respective years, regardless of their changes in time. The URIs have been constructed based on meaningful identifiers (KTS unit codes). A more appropriate situation would probably be a thorough analysis and inventory of statistical unit boundary changes in time and providing separate geometry instances with non-meaningful identifiers (UUIDs). That would mean for example a single geometry with a defined period of validity for a unit which boundary did not change over the years. The quality of geographic data, Statistics Poland had at its disposal, did not allow derivation of boundary changes via spatial analysis (though attempts have been performed). This is mainly due to the fact, that administrative boundary layers provided by the Polish mapping agency were of different quality in different years. Both the precision and methodology of boundary layer creation changed over the years and only recently has the administrative division been harmonized with the cadastral division of the country. The temporal aspect of statistical units geometries was too big an issue to solve within this project, however the issue persists and needs to be addressed in the future.
* **Most linked open data implementations are technically correct,** which is both good and bad news. The upside is that by using existing software or programming components it is nearly impossible to produce incorrect RDF metadata files, regardless of the chosen encoding (e.g. RDF-XML or TTL). The downside is, that most linked open data producing components allow encoding almost anything into triples, so the implementations may not always make sense **semantically**.
* **Linked open data implementations based on Python scripts are easy to amend in the future.** A big advantage in building a linked open data implementation based on Python scripts is their flexibility, which allows easy changes and amendments in the future. Triple creation using Python is based on iterations, so if there is a need (for example) to alter the address of an endpoint, it can be easily done by changing only one variable in the script.
* **RDF vocabulary specifications are easier to interpret with a UML model provided.** This is a somewhat obvious conclusion, but it is worth noting especially for beginners in the field of linked open data. The number and complexity of published vocabularies is already a bit overwhelming and it is growing over time. It is much easier to achieve a good understanding of ontologies if they are represented graphically. A standout example in this case is the DCAT-AP specification, which provides a full UML model of all used classes and properties with clear indication which of these classes and properties are mandatory, recommended or optional. The RDF Data Cube Vocabulary specification also has a simple graphical representation of some of its classes and their relations, which is of great help as well.

1. <http://www.opengeospatial.org/standards/geosparql> [↑](#footnote-ref-1)
2. <https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe> [↑](#footnote-ref-2)