Applying the Semantic Web to The National Map Data

Matthew Wagner[1*], Tanner Fry[1], and Jacques Bourquin[2]

[1] Student contractor to the U.S. Geological Survey, Center of Excellence for Geospatial Information Science
[2] JacoScript, Student contractor to the U.S. Geological Survey, Center of Excellence for Geospatial Information Science

*correspondence: mewagner@contractor.usgs.gov

Keywords: geographic data, linked data, semantic web

Introduction

The Semantic Web was first introduced in 2001 as an idea to create an Internet capable of being fully understood and operated on by machines. This presentation covers the work currently being done by the U.S. Geological Survey (USGS) Center of Excellence for Geospatial Information Science (CEGIS) on applying Semantic Web concepts to USGS topographic data. The system is being used to test two research objectives: 1) the feasibility of the approaches taken by the Semantic Web for geospatial data within the role played by national topographic data for a variety of applications combined with other datasets, and 2) the contribution to building a body of knowledge about system architecture for geospatial ontologies and linked open data.

The current system consists completely of open-source software that is being used to store, convert, process, and link geographical data to the rest of the Linked Open Data (LOD) cloud. This presentation discusses basic workflow and operations of the system, reviews past and current roadblocks, and explores possible future work and directions.

Method

This project focuses on applying Semantic Web concepts to USGS datasets and creating an interactive interface for users to query, search, and find associated data from the LOD cloud. First, data must be converted from their native format to Resource Description Framework (RDF) to be loaded into our system (W3C, 2014). The data are mapped to ontologies that currently exist to ease linking to the LOD cloud. The description of the fields in reference to the data is matched to various fields from pre-existing ontologies including DBpedia, GeoSPARQL, wgs84_pos, and more (DBpedia, 2019; Perry, 2012; Brickley, 2004). Data are mapped to specific predicates via Web-Karma. Web-Karma is an open-source project that was created for transforming, modeling, and publishing data in RDF format (University of Southern California, 2016). Specifically, Web-Karma can be run as an RDF Generation Representational State Transfer (REST) Service to batch convert data. Web-Karma allows users to map data to a Web Ontology Language (OWL) ontology and create an R2RML mapping file that is used by the REST Service to convert data en masse (W3C, 2012; Das, 2012). R2RML

is a language for expressing customized mappings from a relational database to an RDF format.

Both the ontology and mapping file must be created before data can be converted to RDF. Once the conversion information is created, the following workflow is performed.

- Data are downloaded from the USGS The National Map in Esri Shapefile format (U.S. Geological Survey, 2019b; Esri, 1998). Data downloaded in other formats are converted to Esri Shapefile format using the GDAL tool (Open Source Geospatial Foundation, 2019b).
- Data are uploaded to Geoserver (Open Source Geospatial Foundation, 2019a). Geoserver was chosen since it conforms to the Web Feature Service (WFS) Interface Standard (OGC, 2019). Geoserver allows for bulk conversion of data from Shapefile format to RDF.
- Data are converted from Esri Shapefile format to RDF via the Karma RDF Generation REST Service and imported to Apache Marmotta (Apache Software Foundation, 2018). Apache Marmotta is used as a SPARQL endpoint to store, manage, and access the converted data.

Once the data are converted and uploaded to Apache Marmota, they can be accessed, queried, and linked to the LOD cloud via the user interface. The user interface has some unique features including SPARQL support, GeoSPARQL support, visualization of geographical data, a custom SPARQL/GeoSPARQL query builder, and links to the LOD cloud.

The UI has built-in GeoSPARQL queries for some basic functions that allow a user to travel across different datasets without the need for background knowledge. Additionally, SPARQL queries are implemented to visualize the contents of the data graphs (Harris, 2013). Currently, the two functions supported are “Nearby Points” and “Entities Within”. Both functions are implemented via SPARQL queries. This process works by retrieving the entity’s geometry as Well Known Text (WKT), creating a buffer around that point, and returning all the entities whose geometry is contained within that buffer (Figure 1) (Lott, 2015).

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX sf: http://www.opengis.net/ont/sf#
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?name ?lat ?long ?purpose ?geom ?geometry ?dimensions
FROM <http://localhost:8080/marmotta/context/INPUT_NAMESPACE>
WHERE {
        OPTIONAL { ?subject <http://purl.org/dc/elements/1.1/title> ?name . }
        OPTIONAL { ?subject <http://dbpedia.org/ontology/purpose> ?purpose . }
        ?subject <http://www.opengis.net/ont/geosparql#hasGeometry> ?geom .
        ?geom <http://www.opengis.net/ont/geosparql#dimension> ?dimensions .
        ?geom <http://www.opengis.net/ont/geosparql#asGML> ?geometry .
        ?geom <http://www.opengis.net/ont/geosparql#asWKT> ?gWKT .
        BIND(CONCAT("SRID=4326;",STR(?gWKT),"") AS ?strWKT) .
        BIND("SRID=4326;INPUT_GEOMETRY"^^geo:wktLiteral AS ?inputPoint ) .
        BIND( geof:buffer(?inputPoint, .02 , units:degree) AS ?area )
        FILTER( geof:sfWithin(?strWKT, ?area) )
}
```

Figure 1: The GeoSPARQL Query for “Nearby Points” functionality. INPUT_NAMESPACE is replaced with the graph the user wants to search and INPUT_GEOMETRY is replaced with the WKT geometry the user is using to create the buffer.

The “Entities Within” function pulls in the geometry of the selected entity using its Universal Resource Identifier (URI) and compares all the entities within a graph to that entity using the geof:sfWithin function (Figure 2).

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX units: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX sf: <http://www.opengis.net/ont/sf#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?name ?purpose ?dimensions ?geometry ?flWKT
FROM NAMED <http://localhost:8080/marmotta/context/countyorequivalent>
FROM NAMED <http://localhost:8080/marmotta/context/gnis>
WHERE {
        GRAPH ?g {
        <INPUT_URI> geo:hasGeometry ?gWB .
        ?gWB <http://www.opengis.net/ont/geosparql#asWKT> ?geo .
        BIND(CONCAT("SRID=4326;", STR(?geo)) AS ?wbStrWKT) .
        GRAPH ?h {
                ?subject geo:hasGeometry ?gFL .
                OPTIONAL { ?subject <http://purl.org/dc/elements/1.1/title> ?name . }
                OPTIONAL { ?subject <http://dbpedia.org/ontology/purpose> ?purpose . }
                ?gFL geo:asWKT ?flWKT .
                ?gFL geo:asGML ?geometry .
                ?gFL geo:dimension ?dimensions .
                BIND(CONCAT("SRID=4326;",STR(?flWKT),"") AS ?flStrWKT) .
                }
        FILTER (geof:sfWithin(?flStrWKT, ?wbStrWKT) && ?flStrWKT != ?wbStrWKT)
}}
```

Figure 2: The GeoSPARQL Query for “Entities Within” INPUT_URI is replaced with the URI of the entity inside of which the user wants to search.

All geographical data is displayed on a map interface so that users can visualize what entities are being requested (Figure 3). Multiple graphs can be visualized at once. The RDF triples associated with each entity can be found by clicking on the entity and requesting additional information (Figure 4).

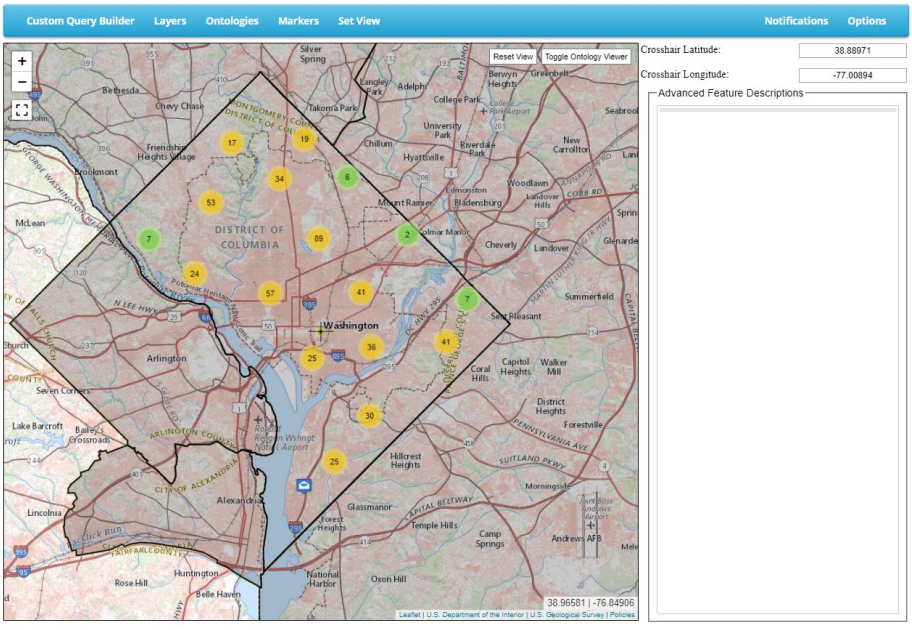

Figure 3: The project UI with the The National Map (TNM) CountiesorEquivalent and TNM Structures datasets displayed.

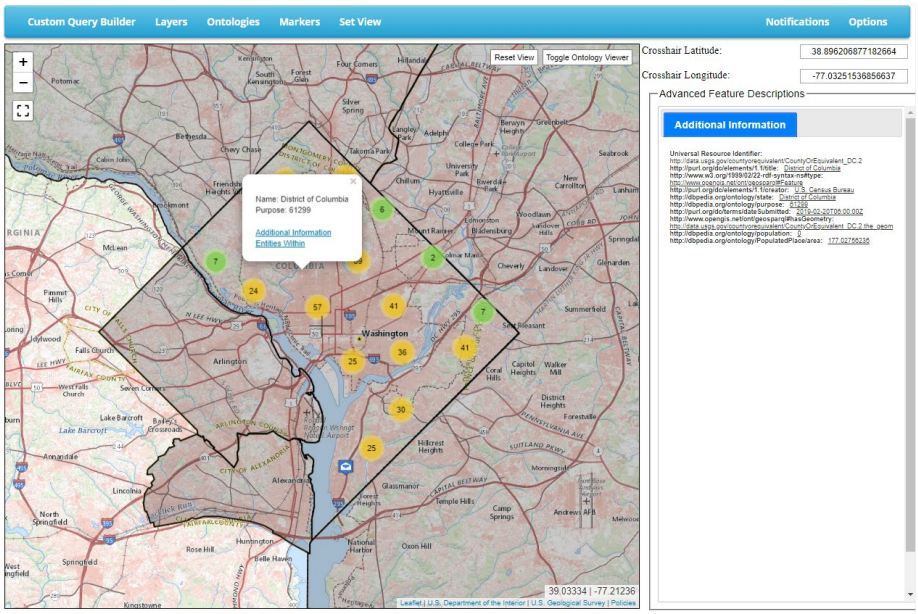

Figure 4: The project UI showing how the RDF triples associated with an entity are displayed.

A custom SPARQL/GeoSPARQL query builder was created that allows users to generate queries or input their user-created queries and run them against the converted datasets (Figure 5). The process works by performing looking-ups on the graphs currently in the datastore and allowing the user to filter the graph based on the predicates of their choosing.

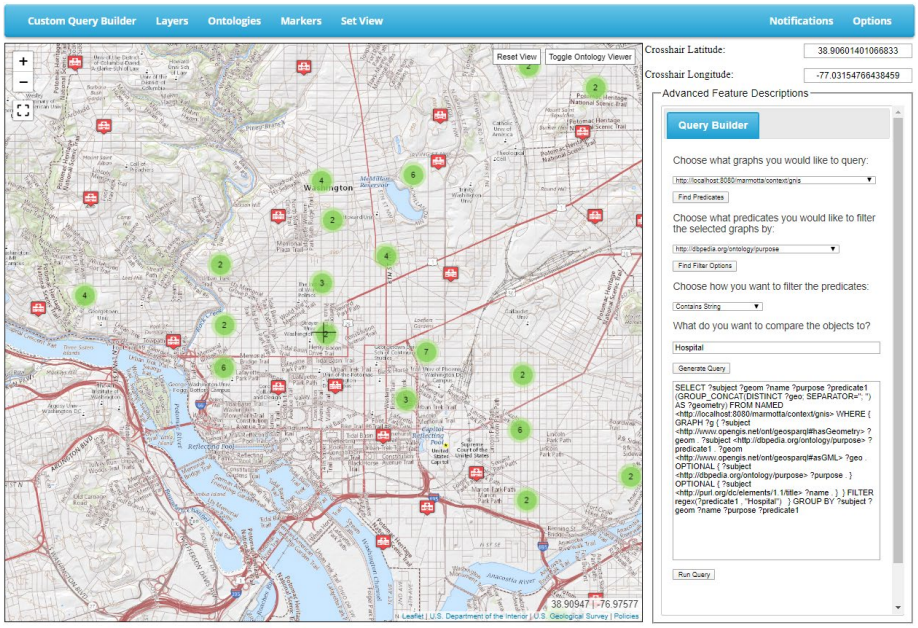

Figure 5: The project UI showing the custom SPARQL query builder and the generated output. Here, the query searched a graph for all objects having a specific string in a specified predicate.

Lastly, the interface links to the LOD cloud. The interface pulls in the data for the requested object and sends a query to other SPARQL endpoints to attempt to find owl:sameAs relationships on-the-fly without the existence of that relationship (Figure 6).

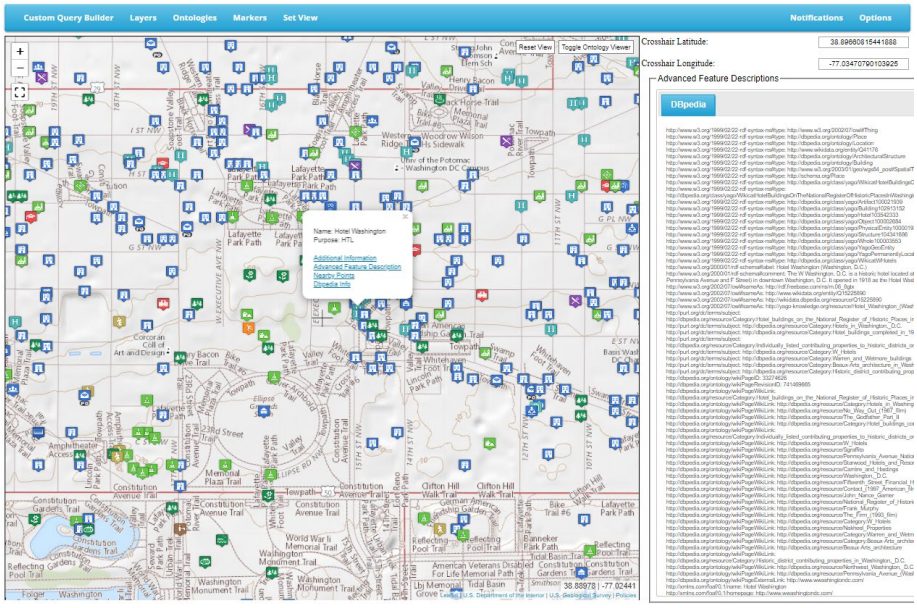

Figure 6: The project UI showing the custom SPARQL query builder and the generated output. Here, the query searched a graph for all objects having a specific string in a specified predicate.

Results

While the project is still ongoing, significant advancements have been made towards bringing the Semantic Web to The National Map (TNM) data. Datasets containing a variety of different geometry types including points, polylines, polygons, and multipolygons have been converted to RDF, visualized, and queried. Additionally, basic on-the-fly entity resolution has been demonstrated by comparing the data of a given entity to entities located in the LOD cloud to learn more about an entity.

```
SELECT *
WHERE {
    ?s foaf:name ?name .
    ?s geo:lat ?lat .
    ?s geo:long ?long .
    ?s ?p ?o
    FILTER (regex(?name, INPUT_NAME) &&
        (?lat >= INPUT_LAT - .001 &&
        ?lat <= INPUT_LAT + .001) &&
        (?long >= INPUT_LONG - .001 &&
        ?long <= INPUT_LONG + .001))
}
LIMIT 100
```

Figure 7: The SPARQL Query for on-the-fly entity resolution. INPUT_NAME is replaced with the name, INPUT_LAT is replaced with the latitude, and INPUT_LONG

is replaced with the longitude of the entity for which the user wants to find additional data.

The project has shown that OWL properties such as owl:sameAs and properties of topological relations found in the GeoSPARQL standard work, as well as topological relations determined between lines and polygons. Additionally, previously known issues with geometry coordinate data storage and processing were improved with additional system capability. The topology of the selected geometry objects was not structured.

Topographic feature types exist in other controlled vocabularies, but the ontologies developed for national topographic features include extensive geometry types, and properties between related entities, whereas vocabularies such as Schema.org (2019) do not. The semantic specification of topographic features and relations enables easy integration with other LOD entities.

Lastly, this project has shown that a semantic technology system for geospatial data can be created from free and open-source software given appropriate server resources.

Discussion and Conclusion

There is a wide variety of different research topics that need to be explored and solved before the current project conforms to Semantic Web standards. This project does not include TNM metadata nor an independent TNM ontology. Instead, the system primarily contains triples that are linked to other ontologies. Additionally, URIs have not been created for most of the dataset. The USGS is currently creating Spec-X, a specification database and application programming interface (API) that houses the metadata and other specification information for TNM data, including data dictionaries and feature definitions accessible in JavaScript Object Notation (JSON) format (U.S. Geological Survey, 2019a). Future work will include creating sample ontologies for TNM datasets from Spec-X, converting the TNM datasets, and integrating it into the system and LOD cloud.

Currently, owl:sameAs relationships are not sufficiently defined between datasets in the LOD cloud. Thus, there is a need to find those relationships without the existence of a ‘seed’ triple. On-the-fly entity resolution has been shown possible between geographical entities. However, most entities fail to find similar entities within LOD datasets due to the immaturity of the comparison operators. Applying concepts such as ontology alignment to allow automated processing and comparisons of entities from different ontologies may prove fruitful as a possible future research direction (Cheatham, 2019). Overall, more work needs to be done to perform accurate on-the-fly entity resolution.

Acknowledgements
This work was performed under U.S. Geological Survey (USGS) contract numbers 140G0219P0002 (Mathew Wagner), 140G0220P0009 (Tanner Fry), and 140G0220P0018 (Jacques Bourquin). Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

References

Apache Software Foundation, 2018, Apache Marmotta: Apache Software Foundation software release, accessed August 2, 2019, at http://marmotta.apache.org/.

Brickley, D., ed., 2004, W3C Semantic Web Interest Group: W3C web page, accessed June 27, 2019, at http://www.w3.org/2003/01/geo/

Cheatham M., Varanka, D., Arauz, F., and Zhou, L., 2019, "Alignment of Surface Water Ontologies: A comparison of manual and automated approaches", Journal of Geographical Systems, 2019. https://doi.org/10.1007/s10109-019-00312-3

Das, S., Sundara, S., and Cyganiak, R., eds., 2012, R2RML— RDB to RDF mapping language: W3C web page, accessed August 2, 2019, at: https://www.w3.org/TR/r2rml/.

DBpedia, 2019, About, Retrieved from http://wikidata.dbpedia.org/about

Esri, 1998, Esri Shapefile Technical Description, accessed November 5, 2019, at http://downloads.esri.com/support/whitepapers/mo_/shapefile.pdf.

Harris, S., and Seaborne, A., eds., 2013, SPARQL 1.1 query language: W3C web page, accessed July 5, 2019, at http://www.w3.org/ TR/ sparql11- query/ .

Lott, R., ed., 2015, Geographic information—Well-known text representation of coordinate reference systems: Open Geospatial Consortium document 12–063r5, 96 p., accessed August 2, 2019, at http://docs.opengeospatial.org/is/12- 063r5/12-063r5.html.

OGC, 2019, Web feature service: OGC web page, accessed July 5, 2019, at https://www.opengeospatial.org/standards/wfs.

Open Source Geospatial Foundation, 2019a, GeoServer: GeoServer software release, accessed August 2, 2019, at http://geoserver.org/.

Open Source Geospatial Foundation, 2019b, GDAL: GDAL software release, accessed August 2, 2019, at https://gdal.org/.

Perry, M., and Herring, J., 2012, OGC GeoSPARQL—A geographic query language for RDF data: Open Geospatial Consortium project document OGC 11–052r4, v. 1.0, 75 p., accessed November 7, 2019, at https://portal.opengeospatial.org/files/?artifact_id=47664.

Schema.org, 2019, Organization of Schemas: Schema.org web page, accessed September 6, 2019, at https://schema.org/ docs/schemas.html.

University of Southern California, 2016, Karma—A data integration tool: University of Southern California web page, accessed August 2, 2019, at http://usc-isi-i2.github.io/karma/.

U.S. Geological Survey, 2019a, Spec-X—Making information accessible: U.S. Geological Survey web page, accessed June 27, 2019, at https://usgs-mrs.cr.usgs.gov/SPECX/treeview/index.

U.S. Geological Survey, 2019b, TNM download (version 1.0): U.S. Geological Survey digital data, accessed July 8, 2019, at https://viewer.nationalmap.gov/basic/.

W3C, 2012, Web Ontology Language (OWL): W3C web page, accessed July 5, 2019, at https://www.w3.org/OWL/.

W3C, 2014, Resource Description Framework (RDF): W3C web page, accessed July 5, 2019, at https://www.w3.org/RDF/.