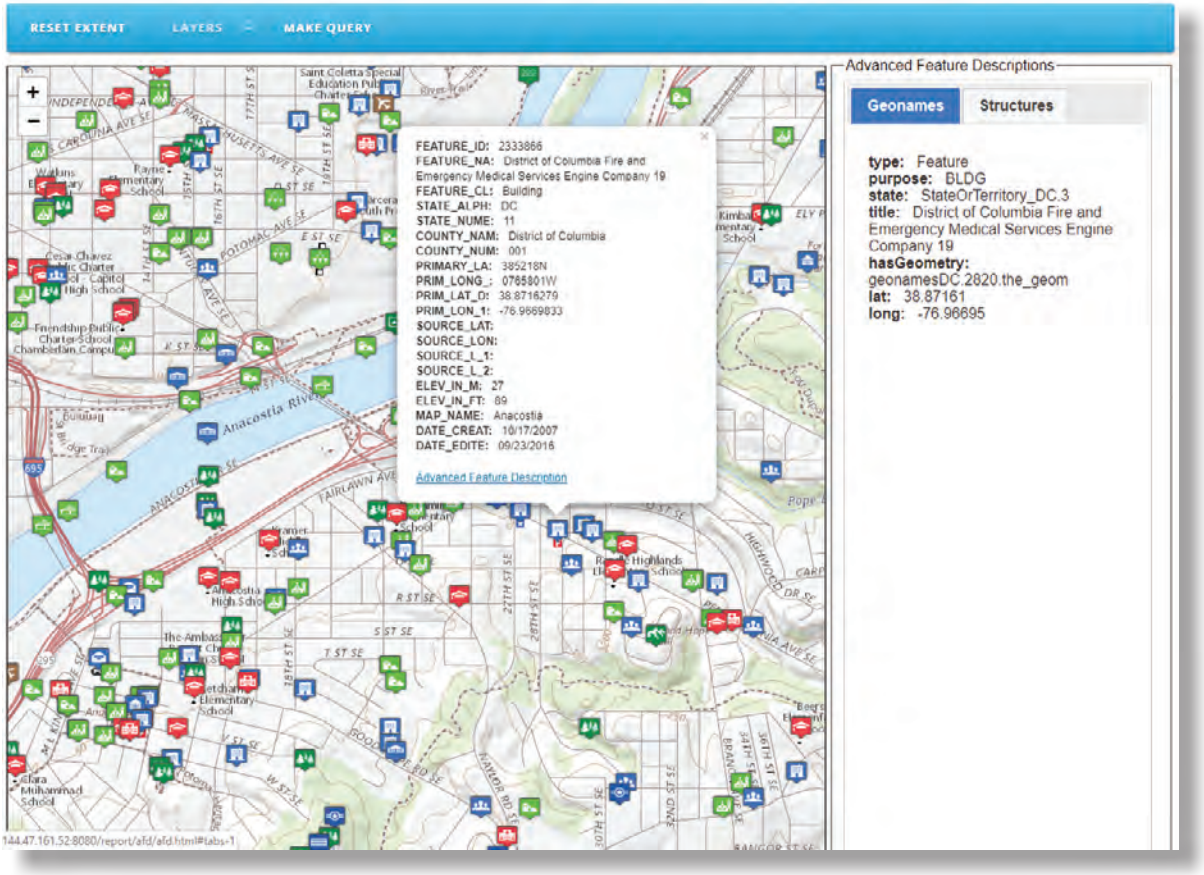
USGS
science for a changing world

A System Design for Implementing Advanced Feature Descriptions for a Map Knowledge Base



Scientific Investigations Report 2019–5148

U.S. Department of the Interior
U.S. Geological Survey

Cover. Maximum scale view showing the initial selection of a feature (fig. 17 in this report).

A System Design for Implementing Advanced Feature Descriptions for a Map Knowledge Base

By Matthew Wagner, Dalia E. Varanka, and E. Lynn Usery

Scientific Investigations Report 2019–5148

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
DAVID BERNHARDT, Secretary

U.S. Geological Survey
James F. Reilly II, Director

U.S. Geological Survey, Reston, Virginia: 2020

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment—visit https://www.usgs.gov or call 1–888–ASK–USGS.

For an overview of USGS information products, including maps, imagery, and publications, visit https://store.usgs.gov/.

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Although this information product, for the most part, is in the public domain, it also may contain copyrighted materials as noted in the text. Permission to reproduce copyrighted items must be secured from the copyright owner.

Suggested citation:
Wagner, M., Varanka, D.E., and Usery, E.L., 2020, A system design for implementing advanced feature descriptions for a map knowledge base: U.S. Geological Survey Scientific Investigations Report 2019–5148, 25 p., https://doi.org/10.3133/sir20195148.

ISSN 2328-0328 (online)

Foreword

The U.S. Geological Survey (USGS) acquires, maintains, integrates, and manages for delivery foundational geospatial information as its mission to provide topographic information for the United States. I have been lucky to have a career at the forefront of the technological changes to advance this mission. The USGS The National Map, includes topographic maps and geographic information systems data for elevation, hydrography, watersheds, geographic names, man-made structures, orthoimagery, government units/boundaries, transportation, and land cover. The end objective is to provide geospatial information to the consumer that is discoverable via search, documented with metadata to assess appropriate use of the data, and then easy to access and use. Semantic technology and linked data and information graphs are the next big change to improve the "delivery" or the ease of use of the geospatial data in a foundational way, much more than adding another attribute or deriving a new product.

Research continues into the use of semantics and ontologies to move the world wide web from human centric to being more readily consumed by machines, which is the most promising way to handle the massive volumes and rich variety of information. The geospatial industry brings a wealth of potential for advancing the consumer's ability to be geospatially aware and connected to their physical surroundings or to forge relationships with locations where they have an interest, be it financial, recreational, scientific, educational, or curiosity. The authors of this paper are working diligently to bring the vast amount of national coverage geospatial information offered by the USGS into a more natural curiosity-driven "follow your nose" navigation model, which they call "advanced feature descriptions."

The future concept that is promoted by this research is to use the topographic map as the user interface to easily access and discover a wide variety of semantically linked data at the feature level such as a stream, lake, or dam. This will allow the consumer to interactively drive the discovery of connections rather than the map designers precomputing and storing connections. This powerful technique of "linked data" can support a diversity of queries from "what are the current beach conditions on a lake" to "what is the average salary of a homeowner with a lakefront view." It may not be obvious to the consumer, but using a linked data solution that does not require precomputing the answers vastly broadens the scope of queries that can be supported.

The promise of semantic technology has great potential to improve the discoverability and data integration potential of The National Map data. It is exciting to see the progress being made in exploring prototype systems to demonstrate the potential and to work out all the technical details. The immense variety of emerging semantic technology tools is one of the challenges of going operational. As this paper adeptly describes, there are more than half a dozen tools involved in this prototype alone, and some substantial barriers remain in going operational for a nationally scoped program.

Semantic technology application solutions are a powerful way to deal with the vast volume of data, the wide variety of data, and even the velocity of data updates that are available. Geospatial relation concepts of "near," "crosses," or "contains" are conceptually simple for humans but daunting for technical solutions that can match performance expectations; that is, be fast enough. The USGS will continue to chip away at the challenges and barriers, and I am looking forward to the day when The National Map is operational on the Semantic Web at a national scope!

Phyllis Altheide
National Geospatial Technical Operations Center,
Associate Director for Innovations

Acknowledgments

We gratefully acknowledge the following named experts for their involvement in the work described in this report. Todd Pehle, Infiniti Consulting Group, LLC, provided an initial design of the system. Nathaniel Davis, former U.S. Geological Survey contractor, integrated original software system components with the server.

From the U.S. Geological Survey, Jennifer Walter facilitated the use of the U.S. Geological Survey Structures data from The National Map; Richard Brown aided with accessing data services from The National Map and other valuable advice; and Michael Speak managed the institutional resources for building the system.

Contents

Figures

Tables

Conversion Factors

International System of Units to U.S. customary units

Multiply	By	To obtain
	Length	
meter (m)	3.281	foot (ft)
meter (m)	1.094	yard (yd)
kilometer (km)	0.6214	mile (mi)

Abbreviations

FCode	feature code
GIS	geographic information system
GML	Geography Markup Language
GNIS	Geographic Names Information System
LD	linked data
lidar	light detection and ranging
LIMES	LInk discovery framework for MEtric Spaces
NSD	National Structures Dataset
OGC	Open Geospatial Consortium
OWL	W3C Web Ontology Language
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
URI	Uniform Resource Identifier
URL	Universal Resource Locator
USGS	U.S. Geological Survey
WFS	Web Feature Server
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

A System Design for Implementing Advanced Feature Descriptions for a Map Knowledge Base

By Matthew Wagner, Dalia E. Varanka, and E. Lynn Usery

Abstract

A prototype system to explore Linked Data that semantically integrates geospatial data in various formats from different publication sources with data from The National Map of the U.S. Geological Survey is presented. The focus is on accessing advanced feature descriptions for data from The National Map with data coreferenced from other sources. The prototype uses Geoserver to access The National Map data, which are converted to Resource Description Framework triples using Karma and stored in the Marmotta triplestore. Marmotta uses a Postgres relational database as a backend for the project and queries to the Marmotta triplestore are converted to structured query language and executed by Postgres. Triples retrieved are linked with same_as relationships to external data sources. The links to these sources provide additional attributes and relationships of the data from The National Map. Visualization of the results is provided using Leaflet and workflows for all parts of the system are defined. A use case for the system is provided to access structures and names information from The National Map for the Washington, D.C., area and link these to Geonames data, with visualization of the graphical and tabular results.

Introduction

Geographic data and information are being developed and delivered at increasing volumes and rates to support new applications, scientific research, and new business models. To support these major transitions in science, business, industry, and society using geospatial data and information, the U.S. Geological Survey (USGS) is researching and developing science, methods, and prototype implementations that allow connection to a geographic feature, attribute and relation connections to additional data and information sources, visualization of retrieved data, and knowledge building through linking data across multiple sources and platforms. This work specifically uses data and connections from The National Map of the USGS as a base to which other data are linked and accessed. This science and development effort specifically supports extensions of USGS geographic data to include semantics and knowledge building through links to other data sources.

Finding, retrieving, and integrating geospatial feature data for a geographic information system (GIS) is time and labor intensive. Information meaning and data relations must be predefined in the design of data formats and the program code before application. This means that when something changes (for example, new features have been created), previously unexchanged information needs to be exchanged, or two programs need to interoperate in a new way, and manual intervention by humans is required. The growth of large databases is making techniques that were developed for traditional information technology less effective because of the volume of data to maintain and publish, the heterogeneous data semantic specifications involved with different user communities, and the increased speed of data update and maintenance schedules.

This transition in data collection, storage, processing, and delivery technology has led to new types of information systems that aim to more efficiently complete these tasks. Semantic technology and linked data (LD) are one potential solution. In many ways, LD extend human-readable web technologies of Hypertext Markup Language (known as HTML) tags and Universal Resource Locator (URL) links (that link text documents), providing a framework that promotes the development of a machine-readable web (that links data sources) that can be traversed, connected, and queried by automated systems. Semantic technology is a set of methods and tools that provides advanced means for categorizing and processing data, as well as for discovering relations within varied datasets, and uses LD. These techniques provide an applied ontology layer to enable machines to automatically search, process, and deliver information graphs. Semantically supported LD are one proposed tool to users who face problems of large database creation, maintenance, and integration such as those of The National Map (U.S. Geological Survey, 2013).

This report describes a prototype system development to explore LD that semantically integrate geospatial data in various formats from different publication sources with data from The National Map of the USGS. The term data integration normally involves combining disparate data within a single unifying framework. The unifying framework for LD is based on declaring certain attributes of geospatial data to be equivalent where they coincide and associating noncoincident properties that enhance the data semantics of the integrated datasets. Such integration is possible using a semantic technology format that is not necessarily the format of the input data

or the final, published output data. The integration format supports the meaning of the data and is easily converted to a technical data model.

Objectives

The project described in this report contributes to a multistage project for which the basic semantic technology configuration is based on the following research questions.

- What technical procedures can enable semantically supported data integration with The National Map? Current The National Map data are isolated and not connected, linked, or semantically connected to other sources.
- Can data from external sources coreferenced with data from The National Map enrich user experiences?
- What methods of generalization and symbolization work with a cartographic user interface design that supports LD?

Coreferences are data that refer and link to the same or related entity. The creation of coreferences has implications not only for delivering The National Map as a LD service, in that coreferences support data integration, but also for the cartographic user interface. The user benefits from coreferences because the interface displays properties they may not have been aware of. These are then directly accessible from the LD graph.

Conceptual Level Summary

The project described in this report is one stage toward the long-term objective to research the concept of a functional map as a knowledge-base component of The National Map (Varanka and Usery, 2018). The core concept for the system is illustrated in figure 1. Data schemes from The National Map and other data stores are used to set the initial parameters. Data are then retrieved and converted to triples to create data graphs if they contain an advanced feature description of another component. The dereferenced data are shown on a cartographic user interface, and users can further query related data by clicking on data shown on the map. The system retrieves the related linked dataset and updates with the linked information. The retrieved data that were linked to the originally selected feature have additional data linked to them. In this way, the user can browse the graph along the LD network.

Previous publications describe aspects of the general conceptual plan such as data storage for map interface access (Baumer and others, 2018), cartographic access to data from The National Map (Powell and Varanka, 2018), data schema designs for The National Map as LD (Usery and Varanka, 2012), and data conversion from The National Map to Resource Description Framework (RDF; Bulen and others, 2011). This central idea for the stage of the project described in this report focuses on linking data. The advanced feature descriptions for the coreferencing process that must precede an LD network were completed. However, users can only get the advanced feature description; browsing data are the next stage of the project.

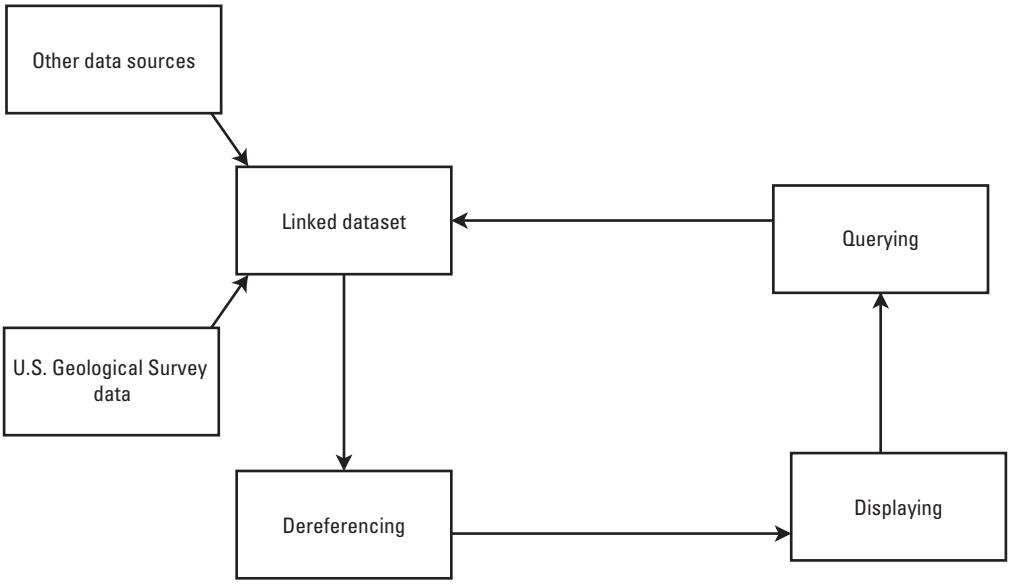


Figure 1. Conceptual diagram summarizing the primary stages of a map as a knowledge base.

The background sections of this report address information on the Semantic Web, which is the framework that supports the technologies used in this project, and The National Map. A description of the main components of the prototype is presented: an overview of the data and software, a detailed description of the system, and discussion of its development. A demonstration of a use case is included in a later section. After a demonstration of the system, discussion and conclusion sections are presented.

Background Knowledge

The Semantic Web

Semantic Web technologies are based primarily on the World Wide Web Consortium (W3C) standards. RDF is a data modeling environment based on triples that take the general form of node-edge-node, or subject-property-object or literal, where the property is a formal logic axiom (W3C, 2014). Each triple resource is identified by a unique Uniform Resource Identifier (URI), called the namespace, that indicates the publisher of the class node or property resource. In this way, each data class or property of each triple can be dereferenced over the internet using the URI, which resolves to a URL (Lewis, 2007). Collections of triples form graphs because a node can have any number of properties that link it to other nodes. This three-part data format is roughly analogous to a simple natural language sentence that is largely intuitive to users, and traversing from one link to another creates logically connected sets of information.

The base URIs used as triple namespaces can be cumbersome to repeat in graph datasets. Namespaces are usually shortened to qualified names that consist of a prefix for the namespace followed by a triple resource fragment, separated by a colon; for example, the namespace for the W3C Web Ontology Language (OWL) is "http://www.w3.org/2002/07/owl#." However, it is normally replaced by the prefix OWL so that a triple resource term from the OWL controlled vocabulary is used as owl:sameAs, for example. Prefixes are used in this project and are presented in this report.

Typically, LD systems are simple data delivery systems. The relations in a LD system are structured by schemas for organizing the logical data interaction, called an ontology. An ontology is a model formalizing real-world concepts into a framework for organizing data. Ontologies are designed and developed using ontology design software that supports OWL (W3C, 2012). The design software usually has a small data storage capacity, typically enough to test models by applying queries against sample data. The final ontology files are then imported to an RDF triplestore database to work with instance data representing individual features. The design software includes one or more reasoners to run on the classes and properties to test the logical integrity of the model. After the reasoner is executed, inferred associations based on the transitive rule will show newly linked sets of data.

The National Map

The National Map consists of eight data layer themes: vector data for boundaries, transportation, structures, and hydrography in Esri Geodatabase formats; point data for geographic names in delimited format; elevation in LAS and LAZ format for lidar (light detection and ranging) data; and orthoimagery and land cover in raster formats distributed as Tagged Image File Format (known as TIFF). The National Hydrography Dataset is a multiresolution database from maps of several scales including 1:100,000-, 1:24,000-, and 1:5,000-scale sources or better, and most recently (as of 2015), derived from 1-meter (m) resolution elevation data from lidar. The USGS elevation data from the 3D Elevation Program are enabled for multiple resolutions with complete U.S. coverage at 30- and 10-m resolutions; partial coverage at a 3-m resolution, and currently (2019) generating 1-m resolution elevation from lidar sources. The National Map supports Open Geospatial Consortium (OGC) services standards for accessibility (U.S. Geological Survey, 2019a).

Data and Software

Only coordinate point-based datasets from The National Map were included for coreferencing with external organization data in this stage of the prototype LD system. These point coordinate data were geospatial features from the National Structures Dataset (NSD) and Geographic Names Information System (GNIS) (U.S. Geological Survey, 2019b). The dataset from an external organization is GeoNames, a geographical database of global scale (Wick and Boutreaux, 2019). The LD graphs derived from the NSD and GNIS conform closely to the attribute tables of the GIS or tabular data (U.S. Geological Survey, 2019c); however, modifications are made to eliminate aspects of the model that were designed for functioning with their respective technology. In the following sections, the original data models and the LD graphs are described, and modifications of the tabular models for the graphs are explained. The third dataset, Geonames.org, required similar modifications.

Data Schema

Two ontology models for coordinate geometry predominate when publishing geospatial data, the W3C Basic Geo ontology and the OGC GeoSPARQL standard (both ontologies were published using the same URI prefix). The W3C ontology has the class geo:SpatialThing with subclass geo:Point, whose datatype properties are geo:lat, geo:long, and geo:alt, for which the values are in decimal degrees (Brickley, 2004). The GeoSPARQL ontology has geo:SpatialObject with subclasses geo:Feature and geo:Geometry, related by the property geosparql:hasGeometry (Perry and Herring, 2012).

The GeoNames project uses geo:Point coordinate classes for geonames:Feature. It would be reasonable to link all graphs to the geo:SpatialThing ontology because all the data are point geometries; however, this was not the preferred implementation because the prototype system will be expanded to coordinate geometries beyond points that require the GeoSPARQL ontology. The data using two geospatial ontologies are coreferenced.

Structures Data from The National Map

The data dictionary for Structures is described in the Spec-X system for standards and specifications (U.S. Geological Survey, 2019d). The feature class Struct_Point, defined as a class of geospatial features representing buildings or other structures as point locations, has a table of attributes that include feature taxonomy and related domains. The feature subtypes of Struct_Point are ontological with subsumption relations and associated three-digit FType codes. Struct_Point subtypes are expanded with child classes called feature domains (feature code [FCode] domains) consisting of five-digit codes, the first three digits of which refer to the parent class. FType and FCode refer to look-up tables values. In addition to feature type look-up table codes, features have string attributes such as names, unique identifiers, and location references. Non-FCode domains, resembling metadata such as Distribution_Policy, Sec_Classification, or OwnerClass, are also look-up tables with codes. The PointLocationType domain adds spatial context to the location point. Three non-spatial tables that address versioning, metadata, and processing were not within the scope of the project and so were not converted to RDF.

The LD graph conforms to the data model as described by the data dictionary for Structures, but the following modifications were made for RDF best practices. The general concept Struct_Type combines a feature and representation as a geometric point. The LD graph separates the feature type from the geometry type. Subtypes of features form the taxonomic hierarchy of the ontology. Class and property labels keep the same terms but are written in CamelCase; forward slashes, for example "College/University," were eliminated with the result "CollegeUniversity." The FCode is an extension or specification of the Struct_Point subtype, so:featureCode is a subproperty of:subType. Some FCodes are ontology instances (for example, US Capital) and so do not fall within the graph hierarchy but within classes.

Geographic Names Information System

The U.S. Board on Geographic Names (BGN) maintains uniform geographic name usage. The GNIS is the official repository of names data for the domestic United States (U.S. Geological Survey, 2019b). The data semantics of each data field are publicly available (U.S. Geological Survey, 2019e). The data model follows the basic format of gazetteers: a unique identification code for an individual, preferred name and others, geographic point coordinates, and topographic feature type classification. In addition to these are some special codes, such as the date of the BGN decision about the data and other Federal codes. When downloaded, the data are formatted as delimited text (.txt or TXT) tables.

GeoNames

GeoNames features are categorized into 1 of 9 feature classes and further subcategorized into 1 of 645 FCodes (GeoNames). Semantics are specified in the GeoNames ontology (Vatant and Wick, 2012). These data are also downloaded as TXT tables; furthermore, GeoNames data are interlinked, meaning that nested entities, such as administrative units within a country, and nearby features are organized within the feature subgraph.

Linksets

The definitions of the three datasets were compared to each other and an equivalent term from widely used vocabularies, such as GeoSPARQL, was selected based on parallels of their semantics (table 1; Tandy and others, 2017). Equivalent classes are identified in rows. Some datasets have subclasses; where other datasets were lacking equivalencies, no values are indicated. When the GNIS data were converted from delimited text to shapefile format, property names were truncated to conform to shapefile limitations with field names. The output field names can be seen in table 1.

The table has four parts corresponding to (1) feature instance coordinates, (2) feature type classes, (3) feature names, and (4) street addresses. These sections are shown in the table from top to bottom. The first section includes feature identification numbers, the geometry type of the feature instance, and the coordinate value for the geometry type. The GeoSPARQL and the Basic Geo ontologies were used. The NSD takes a GIS shape from the Esri field type geometry, so is compatible with the GeoSPARQL ontology, where geometry objects can be expressed as either well-known text or Geography Markup Language (GML). GNIS and GeoNames data points are modeled more similarly to Basic Geo, using separate properties for latitude and longitude in decimal degrees and elevation in decimal meters. GNIS values for points in degree-minute-second format and elevation in feet were not implemented.

The second section of the table reflects the complication of matching structures data, as described in the "Data Schema" section, to the feature type classes of GNIS and GeoNames. The structures data take the form of codes, called FCodes, that refer to a look-up table. Look-up tables are inefficient in graph datasets because the code literals of properties are duplicated across numerous feature instances and using the codes as links to properties introduces an unnecessary extra step for storing and retrieving data. The conversion of FCode tables to triples

Table 1. Sample of equivalencies drawn between data model terms for geospatial feature identification, geometry, type, and structure address.

[GNIS, Geographic Names Information System; URI, Uniform Resource Identifier; --, dataset lacks equivalency]

Structure	GNIS	GeoName	Object class and property link	Prefix URI
Feature identification number, geometry type, and coordinate value				
OBJECTID	FEATURE_ID	GEONAMEID	geosparql:Feature geosparql:hasGeometry	http://www.opengis.net/ont/geosparql#
GNIS_ID	--	--	geosparql:Feature geosparql:hasGeometry	--
SHAPE	the_geom	the_geom	geosparql:Geometry geosparql:asWKT	http://www.opengis.net/ont/geosparql#
--	--	--	geo:Point	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
--	PRIMARY_LA	--	--	--
--	PRIM_LONG_	--	--	--
--	PRIM_LAT_D	LATITUDE	geo:Point, geo:lat	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
--	PRIM_LONG_1	LONGITUDE	geo:Point, geo:lat	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
--	ELEV_IN_M	ELEVATION	geo:Point, geo:alt	http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
--	ELEV_IN_FT	--	--	--
Feature type class				
FTYPE	--	--	--	--
--	FEATURE_CL	FEATCLASS	envo:GeographicFeature	http://purl.obolibrary.org/obo/envo
FCODE	--	FEATCODE	--	http://purl.obolibrary.org/obo/envo
Feature name				
NAME	FEATURE_NA	NAME	gnis:Name	https://data.usgs.gov/gnis
--	--	ASCIINAME	--	--
--	--	ALTNAMES	--	--
Feature street address				
ADDRESS	--	--	schema:address, schema:PostalAddress	https://schema.org/PostalAddress
CITY	--	ADMIN3	schema:City	https://schema.org/City
--	COUNTY_NAM	ADMIN2	schema:addressLocality	https://schema.org/AdministrativeArea
--	COUNTY_NUM	--	--	--
STATE	STATE_ALPH	ADMIN1	schema:State	https://schema.org/State
--	STATE_NAME	--	--	--
ZIPCODE	--	--	schema:postalCode	https://schema.org/postalCode
--	--	COUNTRY	schema:Country	https://schema.org/Country

would facilitate the retrieval of a wider range of terms embedded in the text definitions of codes, though code definitions can reveal contradictory semantic information as well.

The semantics of the values from the third part of the table about feature names were similar and so grouped together under a single base URI for the GNIS data, based on the authoritative status of the U.S. Board of Geographic Names. The fourth part of the table, street addresses, takes the vocabulary of Schema.org, a widely used vocabulary on the internet (Schema.org, 2019).

Data Stores

This section describes data storage and management. The system has two data stores: Geoserver and Apache Marmotta (Open Source Geospatial Foundation, 2019a; Apache Software Foundation, 2018). Geoserver is used to store GML formatted data (OGC, 2019a). Apache Marmotta is used to store RDF formatted data. Apache Marmotta uses PostgreSQL to store the RDF triples (PostgreSQL Global Development Group, 2019).

Data imported into the system must be uploaded to Geoserver. Geoserver can handle a few different input formats. The inputs used for this work are Esri shapefiles created from The National Map for the NSD and the GNIS dataset (Esri 1998). The GeoNames data are downloaded from GeoNames.org (https://www.geonames.org/). The GeoNames and GNIS data are downloaded in text format and converted to shapefile format using the Geospatial Data Abstraction Library (Open Source Geospatial Foundation, 2019b). Using other formats, such as GeoPackage, would preserve the original property names; however, GeoPackage files cannot be used with Geoserver (OGC, 2019b).

Once the data have been imported, Geoserver acts as a Web Feature Server (WFS). A WFS is an OGC interface standard that allows for requests for geographical features across the web using platform-independent calls (OGC, 2019c). WFS acts like a traditional database without any access restrictions. Basic WFS allows for querying and retrieval of features in a variety of formats include comma-separated values (known as CSV), eXtensible Markup Language (XML), and JavaScript Object Notation. A transactional version of WFS allows for creation, deletion, and updating of features. Requests to these data stores can be made via an HTTP request or an XML payload. An example of the HTTP request for a basic WFS is shown below:

https://cartowfs.nationalmap.gov/arcgis/services/structures/MapServer/WFSServer?service=WFS&version=1.0.0&request=GetFeature&typeName=structures:USGS_TNM_Structures&maxFeatures=10&FILTER=<Filter><PropertyIsEqualTo><PropertyName>structures:STATE</PropertyName><Literal>MO</Literal></PropertyIsEqualTo></Filter>

This request was created for the USGS Structures WFS. It can be divided into several parts. The base URL of the USGS Structures WFS data store is copied below:

https://cartowfs.nationalmap.gov/arcgis/services/structures/MapServer/WFSServer?

The service, version, and request are specified as

service=WFS&version=1.0.0&request=GetFeature&

and the data store as:typeName=structures:USGS_TNM_Structures&maxFeatures=10&

With the following, any filters are specified. These filters do not have the same scope as more advanced queries.

FILTER=<Filter><PropertyIsEqualTo><PropertyName>structures:STATE</PropertyName><Literal>MO</Literal></PropertyIsEqualTo></Filter>

Data can be imported into Apache Marmotta in a variety of formats; however, they must be RDF triples rather than the GML data that Geoserver returns. Once the data are in Apache Marmotta, they can be accessed via SPARQL Protocol and RDF Query Language (SPARQL). SPARQL is a query language that is designed for RDF formatted data (Harris and Seaborne, 2013).

Software Description

This section describes the system operation and data conversion processes. The system contains the following tools: Geoserver, LInk discovery framework for MEtric Spaces (LIMES), Karma WFS Plug-In, Karma as a Service, Web Karma, Apache Marmotta, and Leaflet (Agile Knowledge Engineering and Semantic Web, 2018; University of Southern California, 2016; Leaflet, 2019). These tools can be split into two categories: preprocessing tools and query workflow tools. Preprocessing tools are used to set up the data store and the data transformation process. They consist of Geoserver, Web Karma, LIMES, and Apache Marmotta. Query workflow tools are actively used during user requests and consist of Geoserver, Karma as a Service, Karma WFS Plug-In, Apache Marmotta, and Leaflet.

Geoserver is an open-source server written in Java that allows users to share, process, and edit GIS data. Geoserver has a variety of input formats including shapefiles, GeoPackage, PostGIS, and Web Feature Server-NG (Open Source Geospatial Foundation, 2019c; Open Source Geospatial Foundation, 2019d). Filtering is applied on requests to these endpoints following OGC standards (OGC, 2010). Geoserver serves as the primary data store for visualizing data and retrieving advance feature descriptions.

LIMES is a preprocessing tool that is used to create the owl:sameAs relations that are widely used in LD. This tool examines the fields of objects in different datasets to determine if those fields represent the same object. LIMES includes a variety of methods to determine same-as or other ontologically important relations between different entities. These methods include looking for an exact match or a partial match, calculating a Euclidean distance, and more between a data field from

two datasets. These methods can also be combined by using Boolean operators such as and, or, nor, xor, and mathematical operators such as add, subtract, min, max, and, etc. The use of Boolean operators gives users the ability to create indepth entity matching strategies to meet their needs as well as fine tuning matching constraints. More information on LIMES is available in the user guide (Agile Knowledge Engineering and Semantic Web, 2016). If two objects from different datasets do match, LIMES generates a triple with the owl:sameAs property for storage in Apache Marmotta. This triple creates a link between the datasets without having to combine the datasets.

The Karma data tools are divided into three parts: Web Karma, Karma WFS Plug-In, and Karma as a Service.

Web Karma is used as a preprocessing tool to set up the R2RML file. An R2RML file is used for customized data mappings from relational databases to RDF datasets (Das and others, 2012). It is used in our workflow to transform the GML data retrieved from Geoserver to RDF format. Web Karma can be used to transform data via user-created Python functions. This gives the users the ability to manipulate the data in any way they require. Users can add columns, change the data in columns, and even remove columns from the original GML data.

The Karma WFS Plug-In is the endpoint that Apache Marmotta contacts when it pulls data from Geoserver. Apache Marmotta has been specifically set up with known namespaces from the system backend. This means it will search for the data we converted at the Karma WFS Plug-In but it will search for the rest abroad.

Karma as a Service performs the data transformations when the Karma WFS Plug-In is queried. Overall, the set of Karma tools is used to perform the data transformation of the GML data to the specified RDF format.

Apache Marmotta searches for data when requested by the user. It is specifically designed to handle RDF data and perform SPARQL queries. Apache Marmotta also manages SPARQL endpoints to gather data from external sources. Given a specific URL, it either queries the Karma WFS Plug-In or queries another data-stores Semantic Web endpoint.

Leaflet is a JavaScript library used to design the user interface and visualize the requested features. Leaflet is a widely used library for visualizing GIS data. This project uses a plug-in called Leaflet-WFST that allows for easy querying and visualizing of data from Geoserver or another WFS endpoint. Leaflet-WFST creates the WFS queries to send to Geoserver. Only a limited set of filters are enabled. Of the subset of filters that are available, the following were enabled for this prototype:
PropertyIsEqualTo, PropertyIsNotEqualTo, PropertyIsLessThan, PropertyIsGreaterThan, PropertyIsLessThanOrEqualTo, PropertyIsGreaterThanOrEqualTo, and PropertyIsBetween.

Software Interactions

An overview of the way the system and tools interact with one another is shown in figure 2. In the figure, users interact with the Leaflet user interface to request the visualization of a dataset or ask for the system to find additional features from other datasets that belong to that object. Leaflet can interact with either the Leaflet-WFST Plug-In and Geoserver or Apache Marmotta. If the user requests the visualization of features, then Leaflet will send this request to the Leaflet-WFST Plug-In where it creates a WFS request and sends it to Geoserver. Geoserver will return GIS data to the plug-in. The plug-in will create a data layer and send this to the Leaflet interface where it is visualized and updated for the user. This process is described more in depth in the “Visualization Workflow” section.

If the user requests additional features for a specified object, the Leaflet user interface will send the object’s URI to Marmotta. Marmotta looks up the owl:sameAs data that it received from LIMES for any object that represents the same entity. It will then look up the URIs of those objects via Karma as a Service and the Karma Endpoint to retrieve those data. It will return the data back to the Leaflet user interface where the data will be updated.

Overall, the processes done by this LD prototype system are split into three separate workflows: preprocessing, visualization, and advanced feature description. The individual processes are explained and separated in the following three sections.

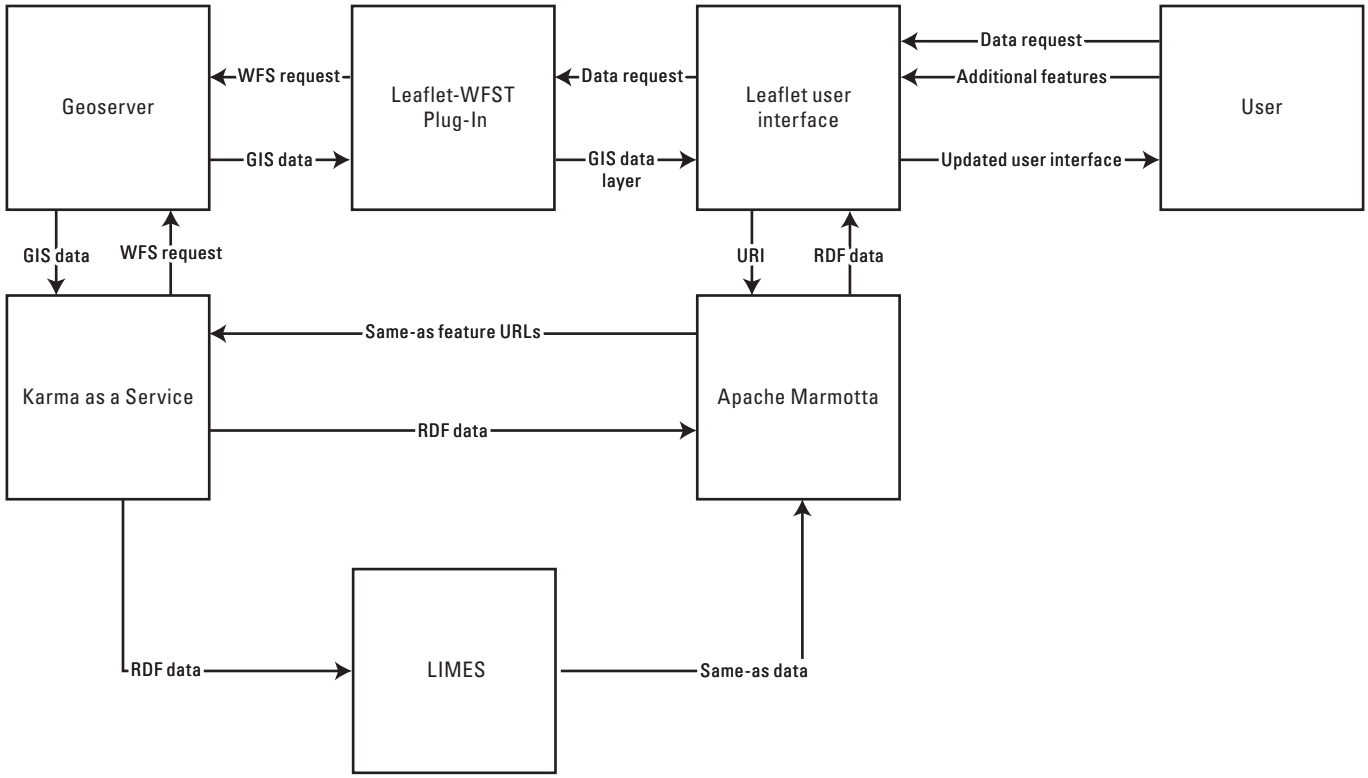


Figure 2. Diagram of the overall advanced feature description system. [WFS, Web Feature Server; GIS, geographic information system; URI, Uniform Resource Identifier; RDF, Resource Description Framework; LIMES, Link discovery framework for MEtric Spaces]

Preprocessing Workflow

The preprocessing workflow includes all the work that must be completed before the tool is operational using Geoserver, Karma, LIMES, and Apache Marmotta. The preprocessing workflow has the following steps (fig. 3):

- Download data from their original data sources and import them into Geoserver.
- Load the data from Geoserver into the Karma WFS Plug-In and design the mapping from GML to RDF. This can include data transformations in the form of Python functions. Publish the data conversion file and the converted data for LIMES.
- Use LIMES to create the same-as relations between objects from two data sources. Output a triples data file with these relations.
- Import the LIMES output into Marmotta with the triples data file so that the system can find the mapping files and the same-as relations for coreferencing requests.

Karma

The Karma web interface is used to create the data mapping from GML to RDF. In the interface, the user loads a sample record from Geoserver. This sample record is used to specify which columns should be accessed and how features should be converted to an RDF equivalent. An example of this can be seen in figure 4.

A user can perform transformations of the data via a user-created Python script. In figure 5, a column named geosparql:wkt was added to the data. This column is the well-known text version of the gml:pos column and can be used with other Semantic Web data sources (Lott, 2015).

Once the data mapping from GML to RDF is finalized, an R2RML file is output. This file is used by Karma to convert any data files that share the exact fields as the original sample record.

LIMES Relation Mapping

LIMES will determine the properties that both datasets have based on their Karma mappings. LIMES compares RDF formatted data. Properties can then be compared using a variety of operators and measures to check if they represent the same object. Images from LIMES and the advanced matching capabilities are shown in figure 6. In the first image (fig. 6*A*),

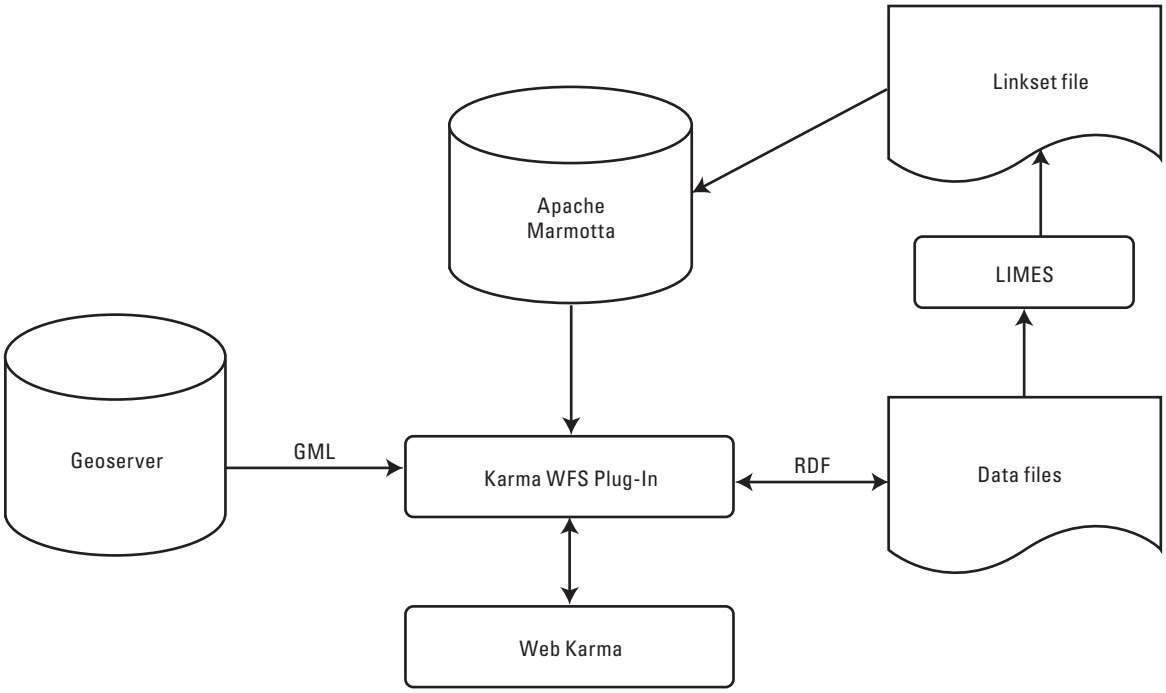


Figure 3. Preprocessing stage software interaction overview. [GML, Geography Markup Language; WFS, Web Feature Server; RDF, Resource Description Framework; LIMES, Link discovery framework for MEtric Spaces]

Figure 4. Interface for selecting data columns for conversion to a Resource Description Framework.

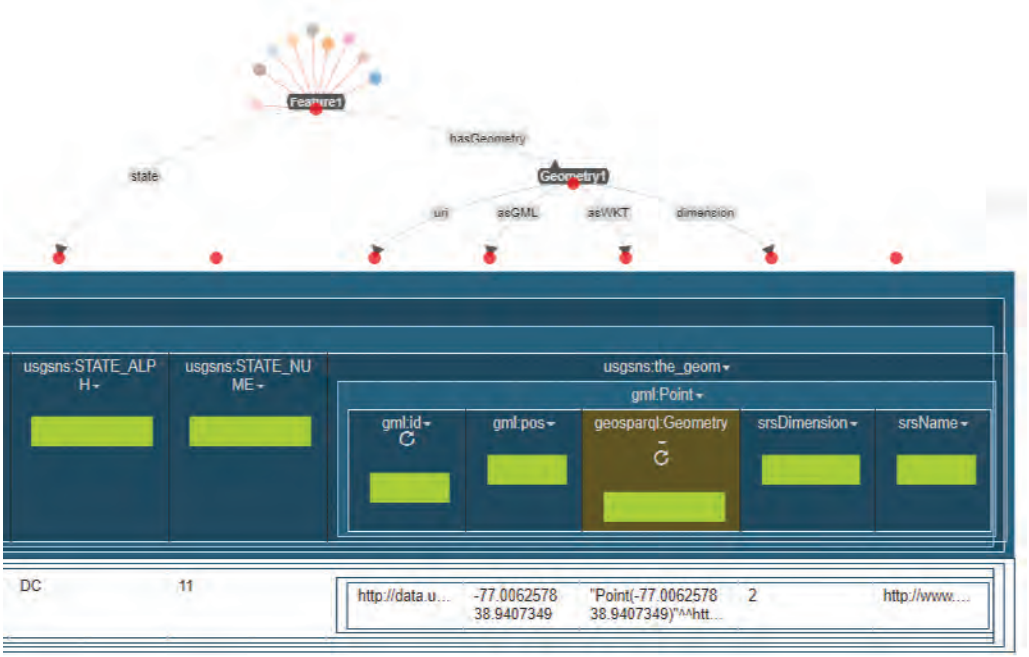


Figure 5. Data modifications using Link discovery framework for MEtric Spaces.

the cosine measure is used to determine if two objects that have the same title or official name are the same; however, this operation allows for some degree of error. For example, an object named Building 1 and another object named Building 2 will be equal because only one character is different. This type of problem is overcome by using multiple attributes from both datasets, which is shown in the second image (fig. 6*B*). Here, the latitude, longitude, and name features of the object are compared. The latitude and longitude use the exact match measure and the "and" Boolean operation is used to combine the comparisons. This allows for increased accuracy for the entity comparisons.

This tool creates an output list of tuples matching the entities from the datasets that is then uploaded to Marmotta. These tuples create data linking between the entities. They are in RDF format and use the W3C standard owl:sameAs relation. An example from Marmotta is shown in figure 7.

After the LIMES data are uploaded to Marmotta, the system is ready to operate. Now that the preprocessing workflow has been explained, a description of the visualization workflow follows.

Visualization Workflow

The visualization workflow includes the tasks completed by Geoserver, Leaflet, and the Leaflet-WFST Plug-In to visualize any data that are requested by the user. The visualization workflow has the following steps:

- The user specifies a dataset request via the Leaflet user interface. The request can be for a complete dataset or can include a filter to select a subset of the dataset.
- The Leaflet-WFST Plug-In creates a WFS query, which is sent to Geoserver.
- Geoserver receives this request and gathers the relevant data that match the feature count and filter. Geoserver sends these data back to the Leaflet user interface in the form of a Geoserver JavaScript Object Notation (or GeoJSON).
- The user interface determines what the data represent (that is, church, school, hospital), then visualizes the data using Leaflet layers and the clustering plug-in.

The visualization workflow process is shown in figure 8.

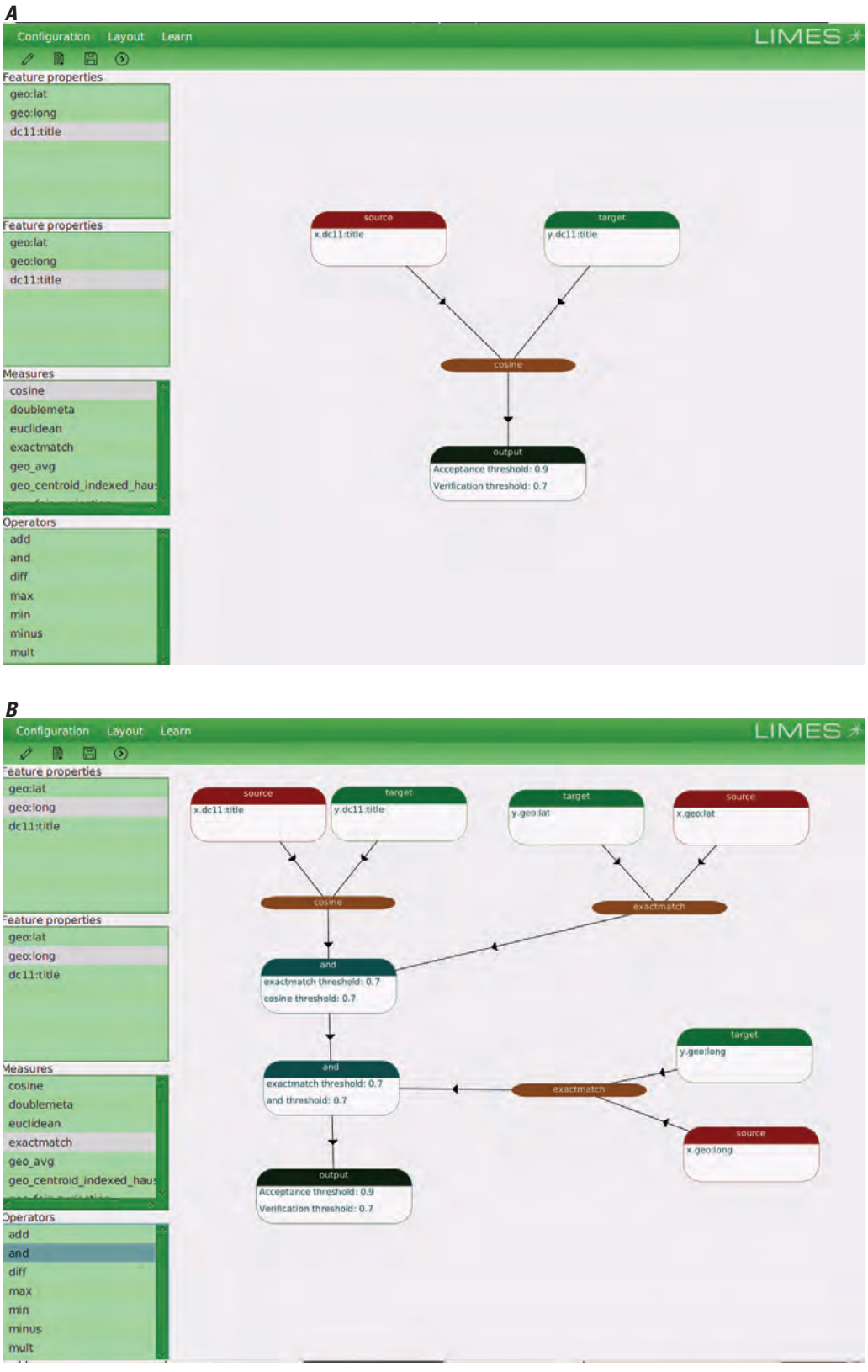


Figure 6. Link discovery framework for MEtric Spaces interface for mapping relations between datasets. *A*, interface showing cosine measure of one attribute; *B*, interface using multiple measures on multiple attributes.

http://data.usgs.gov/gnis/gnis_DC.105	
http://www.w3.org/2002/07/owl#sameAs	http://data.usgs.gov/geonames/geonamesDC.1754
http://data.usgs.gov/gnis/gnis_DC.1677	
http://www.w3.org/2002/07/owl#sameAs	http://data.usgs.gov/geonames/geonamesDC.1481
http://data.usgs.gov/gnis/gnis_DC.1699	
http://www.w3.org/2002/07/owl#sameAs	http://data.usgs.gov/geonames/geonamesDC.1340
http://data.usgs.gov/gnis/gnis_DC.1731	
http://www.w3.org/2002/07/owl#sameAs	http://data.usgs.gov/geonames/geonamesDC.441
http://data.usgs.gov/gnis/gnis_DC.1900	
http://www.w3.org/2002/07/owl#sameAs	http://data.usgs.gov/geonames/geonamesDC.194

Figure 7. Output tuples linking Resource Description Framework resources.

The figure 9 image was taken directly from the screen of the LD prototype. The user selected data from the Geonames.org dataset for Washington, D.C. Note that buildings have different visual representations based on the structure they represent. These representations are based on their feature information and are discussed more in the "Displaying Layer Data" section.

Leaflet Map Display Properties

Leaflet uses unique layers to present information to the user. These layers can be either a base map that sits at the bottom level and is typically used for background images, an overlay map that sits at a specified view level and is typically used for informative displays, or a set of geometric objects that are placed on top of the other layers. The top layer can be made up of many types of data such as popups or markers representing point data, raster layers such as tileLayers and other overlays, vector datasets shown as polylines or polygons, and grouping layers that combine popups and markers to increase performance. All these types of layers are used in conjunction to make an application provide the information requested when examining the data in an easy-to-view format. This application uses the multiscale base map for the USGS, called USGSTopo, as the primary backdrop using a raster tile layer that portrays information from The National Map (U.S. Geological Survey, 2019f). All data that are requested by the user to be visualized are grouped together in a clustering layer and shown on top of the USGSTopo backdrop.

Query results are stored in a nested data structure in the JavaScript Object Notation for Linked Data file format. The file is iterated through, and the data and coordinate information for each entity are extracted individually. This information is bound to a marker. Markers use a latlng object, which is a pair of latitude and longitude attributes, and a custom icon to plot the information on the map. The information bound to a marker will only be shown when requested. These requests take the form of a user clicking on a marker. Leaflet takes all the individual points and gathers them into a clustering layer, which generalizes the data into one layer rather than showing all data points at all zoom levels. Using these clusters allows large amounts of data to be visible without an effect on performance. This allows users to zoom in or out and pan around the map interface without the internet browser crashing. The clustering layer is shown in figure 10. Some points are still shown individually if they are not close enough to any other cluster head to be included.

Displaying Layer Data

Each feature is represented by a specific symbol based on its feature type. Some examples can be seen in table 2. The table highlights the heterogeneous nature of the different datasets; for example, all three datasets contain a different representation of the semantic definition of a hospital. The symbols used to describe a feature are also the same across several feature identifications; for example, School: Elementary, School: Middle School, and School: High School from the USGS Structures datasets are represented by the same symbol because, on a semantic level, they represent

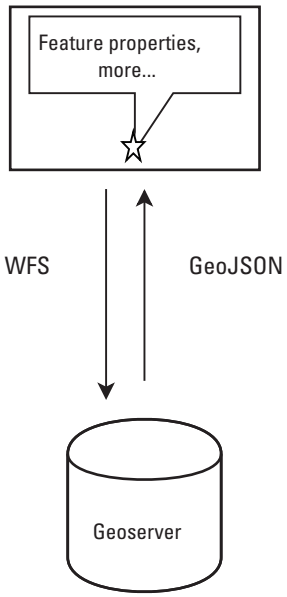


Figure 8. Initial visualization workflow. [WFS, Web Feature Server; GeoJSON, Geoserver JavaScript Object Notation]

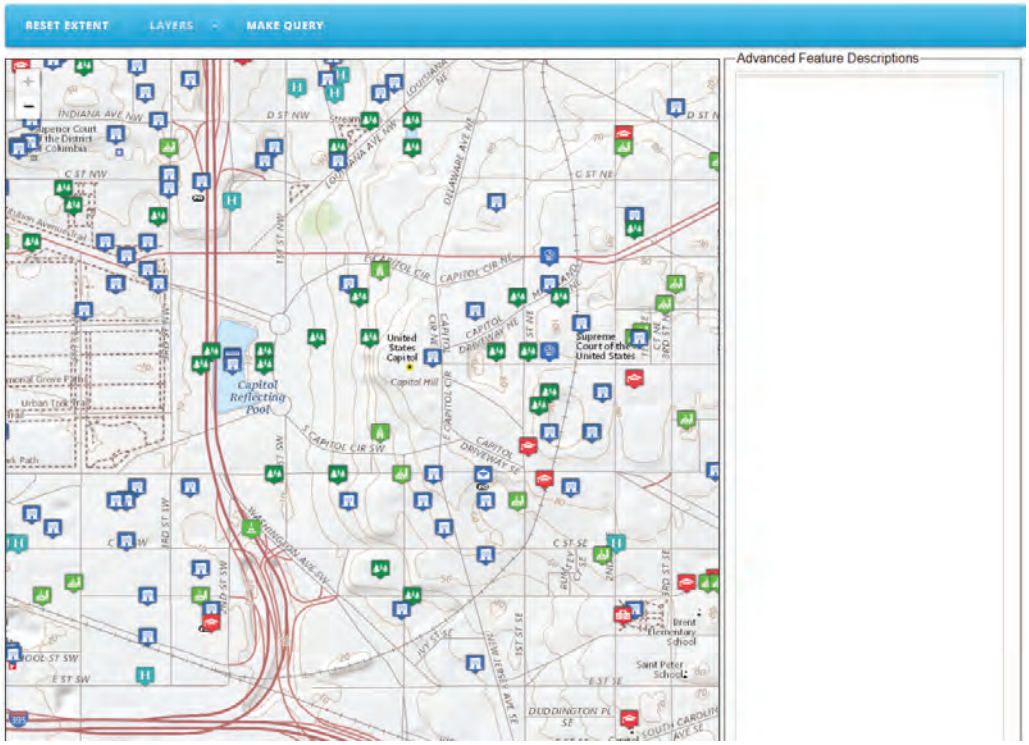


Figure 9. Visualization of the initial retrieved datasets.

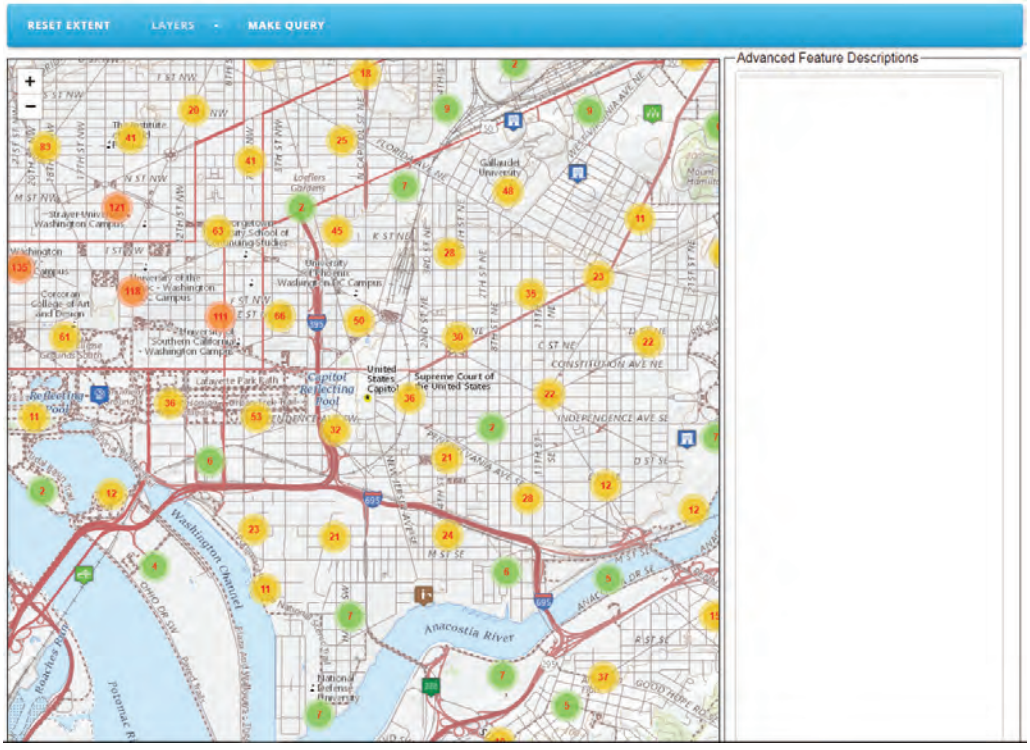


Figure 10. Visualization of clustering on the user interface.

Table 2. Example entries from the U.S. Geological Survey (USGS) Geographic Names Information System (GNIS) map symbol library.

Symbol name	Symbol image	Data representation
Hospital		USGS GNIS: Hospital Geonames.org: HSP, CTRM USGS Structures: Hospital/Medical Center
Populated place		USGS GNIS: Populated Place Geonames.org: PPL, PPLX
Post office		USGS GNIS: Post Office Geonames.org: PO
School		USGS GNIS: School USGS Structures: School: Elementary, School: Middle School, School: High School, College/University, Technical/Trade School Geonames.org: SCH, ITTR, STNB

locations of learning. The granularity of the feature definitions is also inferred in the table. The USGS GNIS dataset does not distinguish between different levels of schools. Instead, all levels contain the same feature attribute.

The feature attribute and symbol representation have no effect on the clustering function. The clustering function takes individual points and groups them together by using a greedy clustering algorithm. The algorithm will select a subset of points in the currently viewable area as cluster heads based on the distance apart and the total number of points that are viewable. Then, it will group the rest of the points with these cluster heads using a greedy approach to assign a cluster head based on proximity. Once the algorithm is complete, the cluster is shown as the position of the cluster head and a display with the number of points in that cluster. When a cluster head is hovered over, a polygon showing the area that the points are in appears (fig. 11). This polygon was generated using the quick hull algorithm to create a convex hull for the cluster. The system only updates the clustering layer once when the user stops zooming or panning because updating the clusters can become performance intensive if called too many times.

As the map is enlarged, clusters begin to break into subplots and center among the collection of points in that area. At a specified zoom (currently set at the maximum zoom level), Leaflet shows the marker icons instead of the clusters so that the user knows more information about individual points.

Hovering over an icon will display a tooltip with that point's feature class so that the user knows what each icon represents (fig. 12A). These feature classes are different for the different data sources. Clicking a marker will display a popup of relevant information regarding that point along with a link for the advanced feature description for that entity (fig. 12B). Clicking on the link will display relevant information from other data sources concerning that point. As the point is used as a subject for queries into other relevant datasets using the linking files discussed with LIMES, new tabs begin to populate the space with new information about the selection. This process for retrieving the advanced feature description data is described in the next section.

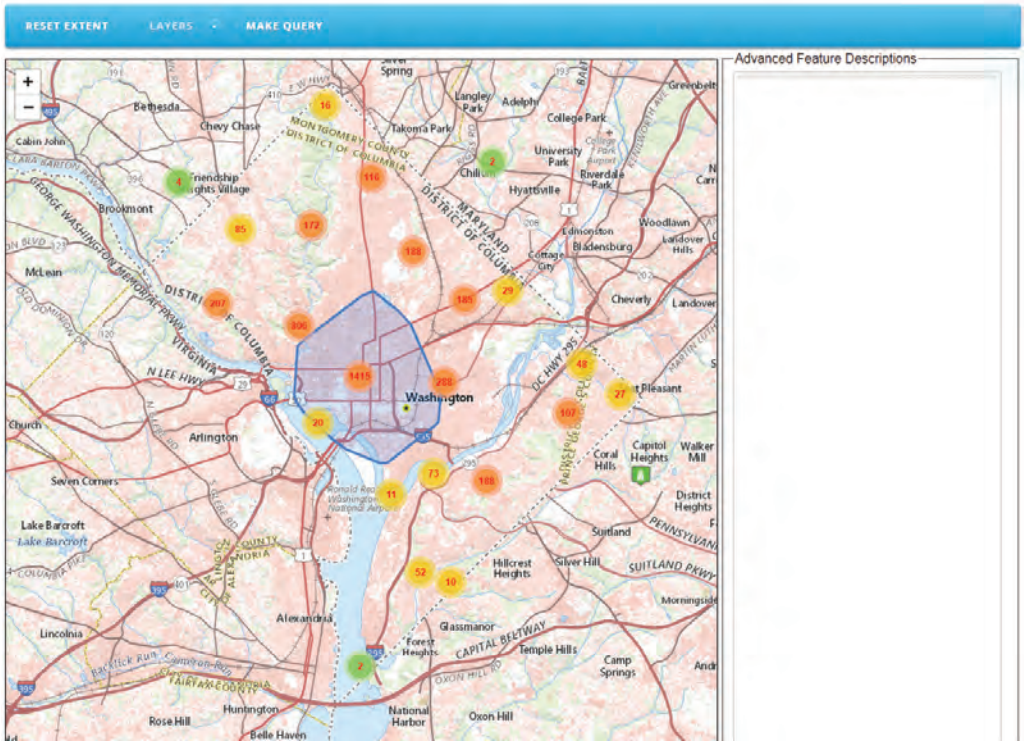


Figure 11. Visualization of a quick hull polygon on the user interface.

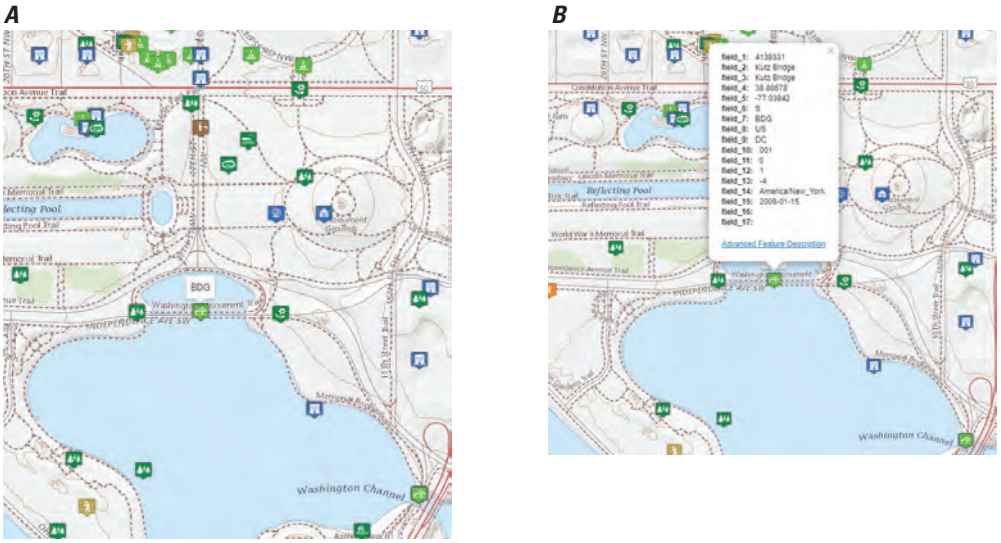


Figure 12. Visualization of Leaflet layer data. A, tooltip containing the feature class; B, popup with the entity attributes.

Advanced Feature Description Workflow

The advanced feature description workflow includes the functions completed to retrieve any additional features for an object that the user specifies (fig. 13). This workflow includes Leaflet, Marmotta, Karma, and Geoserver.

The advanced feature description workflow has the following steps:

- The user specifies an object to request via the Leaflet user interface. Leaflet gets the object URI and creates a Marmotta SPARQL query. The query is designed to look for triples that contain the URI specified in the request. This query gets forwarded to Marmotta.
- Marmotta receives this query. It queries the database for any URIs that have a same-as relation with the selected URI. It decomposes the additional URIs to determine the endpoint that needs to be queried. Any endpoints starting with data.usgs.gov will be forwarded to the Karma WFS Plug-In. Any requests sent to Karma will use Karma as a Service and will forward the request there. Additionally, any other prefixes will be forwarded to the appropriate SPARQL endpoints. The other endpoints will return the records in RDF format.
- Karma takes this request and sends a request for data to Geoserver based on the WFS part of the request.
- Geoserver takes this request, forwards this to the external USGS WFS endpoint, and gets the results for the query. It then returns these results to Karma in the form of GML data.
- Karma takes the GML data and transforms them based on the predefined mapping. Karma then returns these data to Marmotta.
- Marmotta receives these data and forwards them to the user interface.
- The user interface visualizes the requested data or gives the data in a human-readable format for the users.

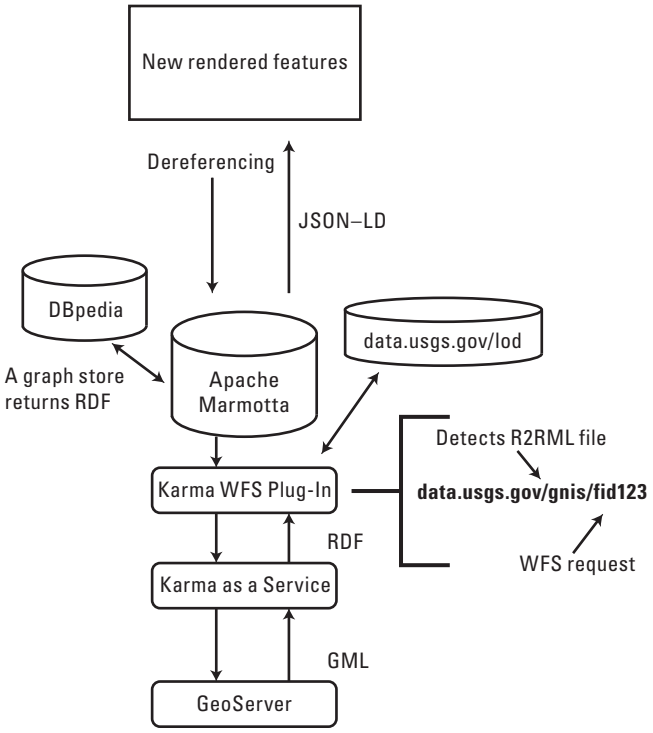


Figure 13. Advanced feature description workflow. The area shown in bold refers to the meaning behind the Uniform Resource Identifiers in our system, as described in further detail in the next section. [JSON–LD, JavaScript Object Notation for Linked Data; RDF, Resource Description Framework; WFS, Web Feature Server; GML, Geography Markup Language]

Use and Querying of a Uniform Resource Identifier

URIs are important when creating any linked dataset. An example of a URI that is generated by and used within the LD prototype is shown in figure 14.

The URI is generated via Karma as a Service during the data conversion process. It is primarily used to tell our coreferencing workflow how to retrieve and convert the data. The URI can be divided into three parts: the namespace, the data conversion identifier, and the Geoserver data URI. The namespace is the general namespace chosen for the data. This information tells Marmotta whether to query the Karma-as-a-Service endpoint or another endpoint, for example, DBpedia. The second part is the data conversion identifier. It tells Karma as a Service which mapping file to use when converting the data from GML to RDF. This is important because the data will not be converted if the wrong mapping file is used. The last part tells Karma as a Service which URI to look up and return from Geoserver.

When a user requests an advanced feature description for an entity, two SPARQL queries are built. The first query using the URI as the subject gathers data that point outward from the URI (query 1, shown below).

1. PREFIX owl: <http://www.w3.org/2002/07/owl#>

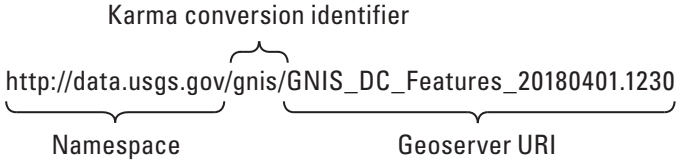


Figure 14. Linked data prototype Uniform Resource Identifier (URI) breakdown.

SELECT DISTINCT ?coref ?dsuri

WHERE { { GRAPH ?dsuri { ?coref owl:sameAs <entity_URI> . } } }

The second query using the URI as the object, gathers data that point inward toward the URI (query 2, shown below).

2. PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?coref ?dsuri

WHERE { { GRAPH ?dsuri { <entity_URI> owl:sameAs ?coref . } } }

In these queries, the entity_URI is the URI for the specific entity that is being queried. The only difference between them is the ordering of the requested RDF triple.

After these queries are sent to Marmotta and the entity that has the owl:sameAs relation is determined, the URIs of the entities who describe the same entity as the current entity are returned. From here, two more queries are created to gather the additional entities information (queries 3 and 4, shown below).

3. PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?coref ?dsuri

WHERE { { GRAPH ?dsuri { ?coref owl:sameAs <entity_URI> . } } }

FILTER(?coref != <filter_URI>) . }

4. PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?coref ?dsuri

WHERE { { GRAPH ?dsuri { <entity_URI> owl:sameAs ?coref . } } }

FILTER(?coref != <filter_URI>) . }

These queries will gather the attributes for the additional entities while filtering out the data of the current entity. Like queries 1 and 2, the only major difference is the ordering of the requested RDF triples. Once the results are returned from Marmotta, Leaflet will create a tab on the right side of the user interface.

Generation and Display of Advanced Feature Descriptions Tabs

As the URIs are dereferenced and feature information is gathered, tabs are created on the right side of the user interface. Each tab is labeled by the dataset the advanced feature description is being taken from (fig. 15). Tabs contain the attribute information for the coreferenced entities. This information is shown when the tabs are clicked by the user.

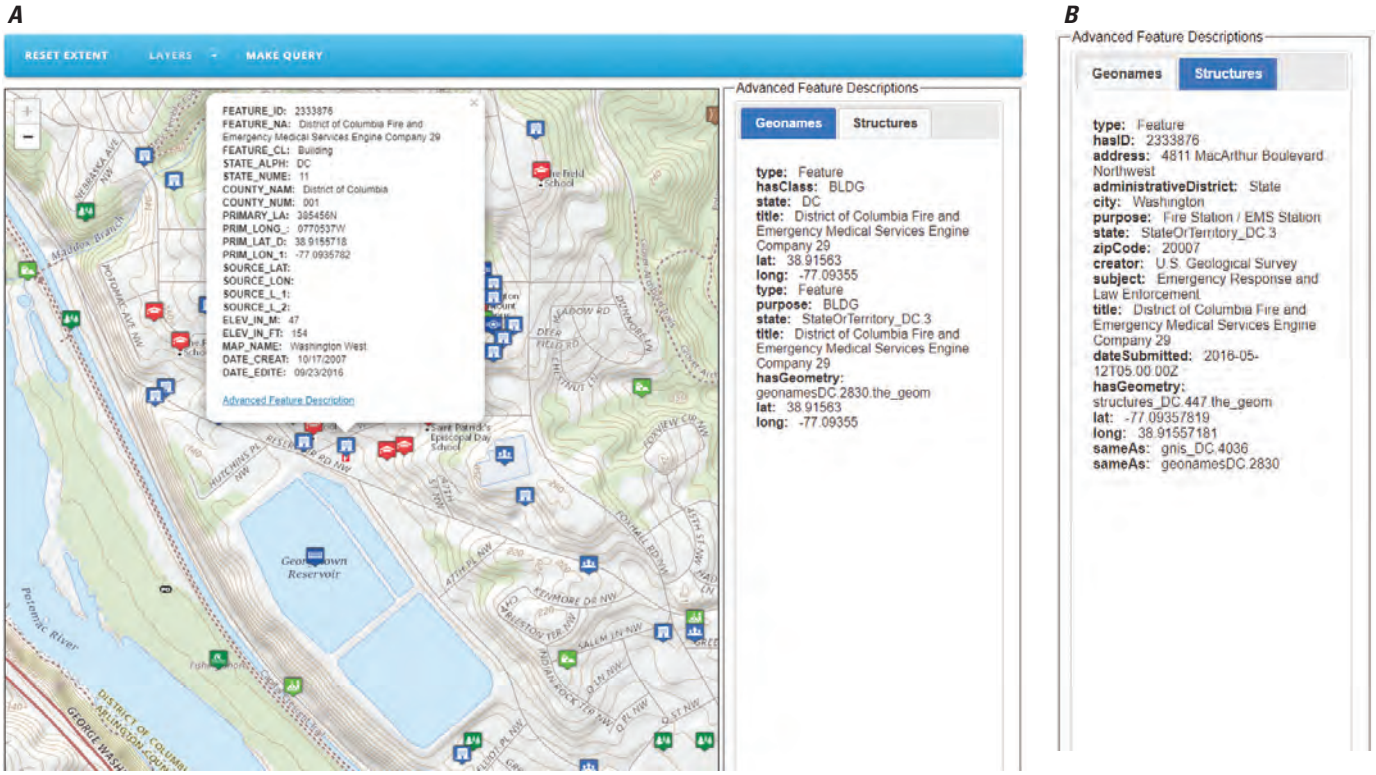


Figure 15. Visualization of the tab generation for a point return. *A*, the Geonames owl:sameAs feature; *B*, the U.S. Geological Survey Structures owl:sameAs feature.

Example of a System Use Case

Data for Washington, D.C., from the USGS Structures, USGS GNIS Names, and Geonames.org data were used for this example. The USGS Structures dataset was loaded from a shapefile into Geoserver. The data can be downloaded from the USGS The National Map data download site (U.S. Geological Survey, 2019g). The GNIS Names dataset was also downloaded from the data download site; however, it is only available in text format and the Geospatial Data Abstraction Library was used to convert it into shapefile format, which was uploaded to Geoserver. Similarly, this was also done for Geonames.org data because the data are only available in text format; however, Geonames.org data were downloaded from their website.

Karma Mapping

Three primary attributes were used to compare entities from separate datasets. These attributes include latitude, longitude, and title. If the data did not include a separate field for longitude and latitude (for example, the GNIS Names data), then a separate data field was created for these via the Python transformation function in Karma. The W3C Basic Geo terms geo:lat and geo:long were used as the new RDF type for the latitude and longitude, respectively. The titles were labeled as a dc11:title in the RDF format.

LIMES Relation Mapping

We used the owl:sameAs relation mapping that was previously shown to link the entities from different datasets. We used the mapping relation from the second example in figure 6 to compare entities. We determined that the Structures dataset had additional features in the GNIS Names and Geonames.org datasets; however, this dataset only contains about 500 entities. Thus, most of the entities in the GNIS Names and Geonames.org datasets do not have an equivalent entity in the Structures dataset.

Geoserver Data Retrieval

The first query shows the retrieval of all the data from the USGS GNIS dataset for Washington, D.C. (fig. 16). Compared to the other dataset, this is a small dataset and includes only 514 entities.

The results from when an advanced feature description request is executed on an entity from the USGS Structures dataset are shown in figure 17. The additional features are on the right-hand panel with tabs noting the source dataset. In figure 17, notice that the data on the right are different from the original dataset. The panel also includes the URI that was used in the additional dataset, allowing the user to look up the same entity in the other dataset.

The form for filtering the USGS GNIS Names dataset is shown in figure 18. In figure 18, the PropertyIsEqualTo filter is selected. This is one of multiple OGC filters currently

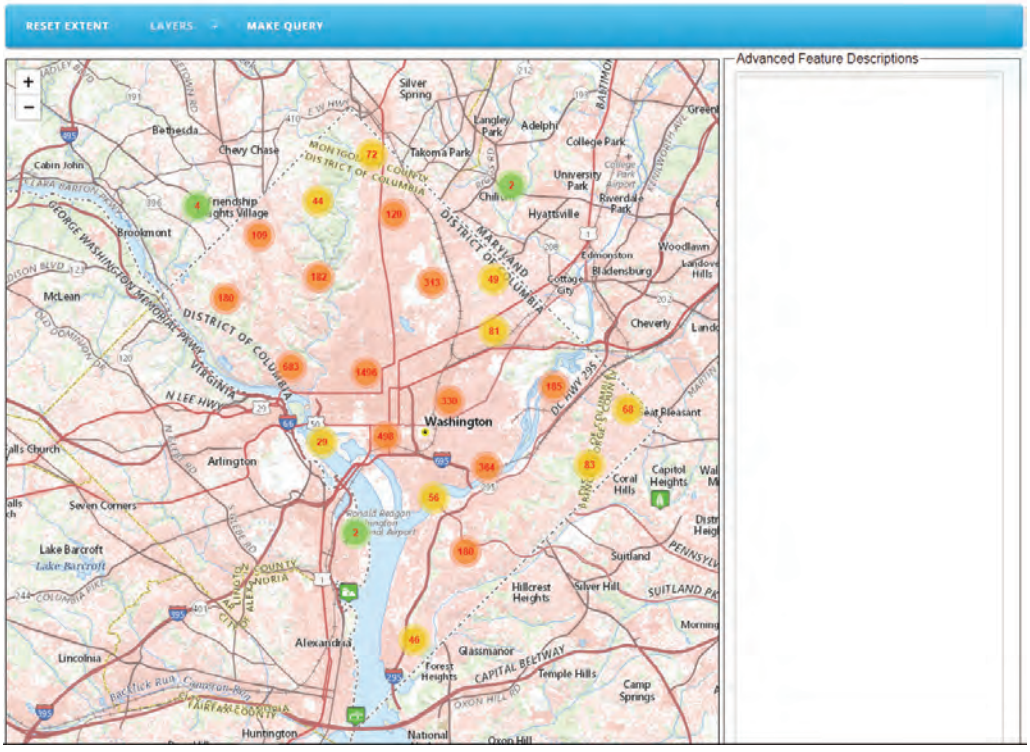


Figure 16. Initial visualization screen showing clustered portrayal.

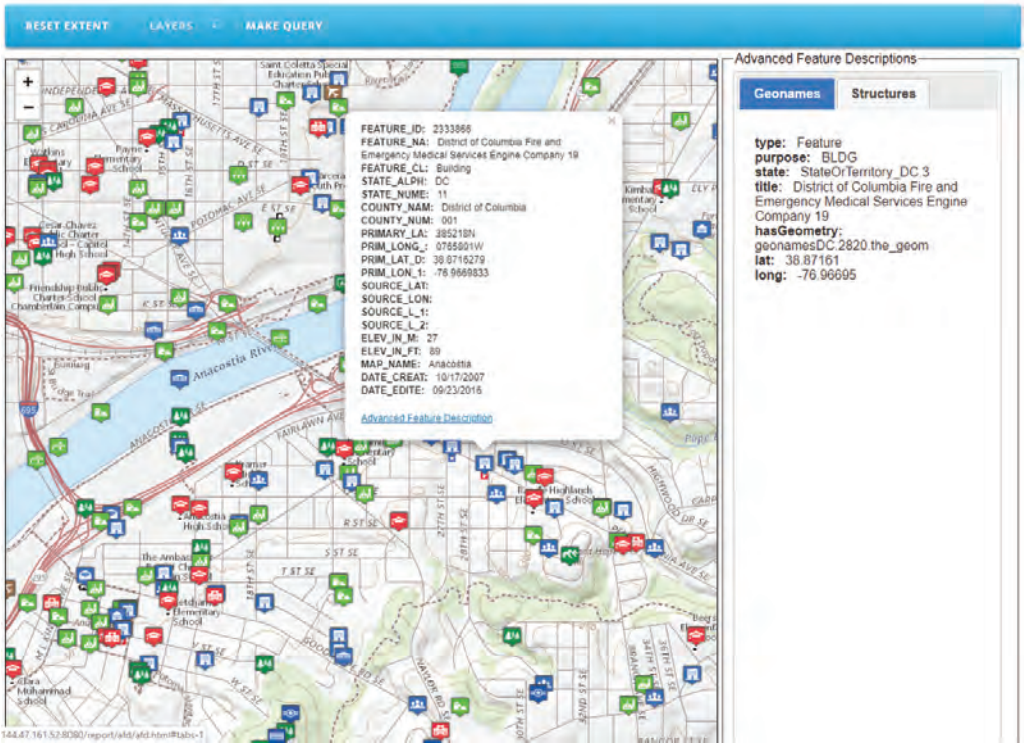


Figure 17. Maximum scale view showing the initial selection of a feature.

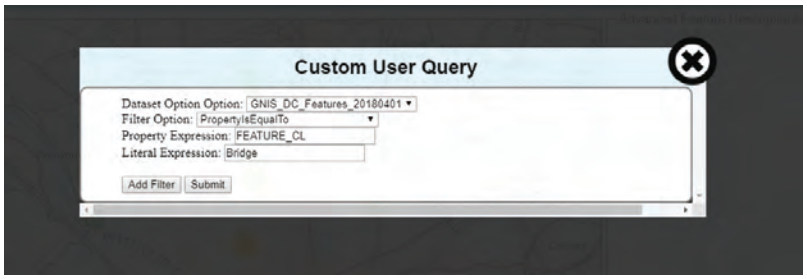


Figure 18. Leaflet Transactional Web Feature Server filter interface.

implemented in the Leaflet-WFST Plug-In (OGC, 2017). The advanced feature descriptions tool currently supports unlimited filters that use the Boolean operator "and" together. This means that any returned data match every filter.

The results of applying the filter from the dataset are shown in figure 19. The Leaflet-WFST will create a filter that matches the OGC standard filter and attaches it to the WFS request it will send to Geoserver (Vretanos, 2010). To use the filter, the user must already know the field name and the literal expression for which they are searching.

The example filter returned geospatial data from other datasets that were the same feature as the one selected by the user. Now that the capabilities of the system have been reviewed, a discussion of the tool and resulting insights is provided in the next section.

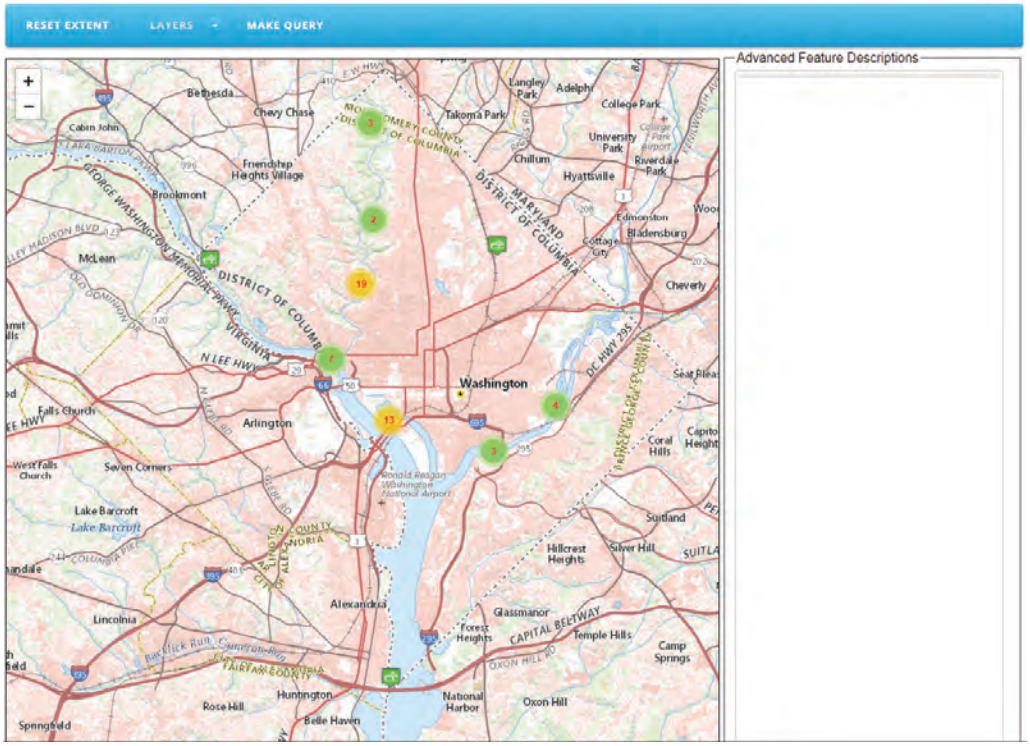


Figure 19. Query results.

Discussion

A summary of the overall technical approach and an evaluation of the strengths and weaknesses of the system are discussed in this section. Conclusions about the challenges and potential alternatives for linking coreferences for LD follow the summary and evaluation. This section concludes with a statement of the significance of the results.

Summary

The previous section described the technical aspects of producing coreferenced advanced feature descriptions of LD with The National Map. The study was designed to test if linking these data is feasible and if it would benefit the user experience and to identify the implications for a LD map user interface. Our results indicate that techniques are compatible with topographic data from The National Map. The preprocessing workflow converted data schemas in multiple formats and coreferences were established between data schemas of subject matter common to multiple datasets. The map interface represented semantic properties of geospatial features.

Strengths and Weaknesses

There are three major strengths to this project. The first strength is the ability to customize the data mapping from GML to RDF. The Karma tool allows users to specify the conversion for any dataset that is in Geoserver, which allows users to create a standard base mapping. Secondly, most geographic datasets can be added to Karma if they can be converted into a shapefile, allowing users to easily visualize any datasets they would like. Lastly, users can filter data, which allows viewing limited sets of the data that are loaded into Geoserver.

There are three weaknesses of this project. The greatest weakness is the manual work required to set up the system. The preprocessing workflow must be completed for every dataset. Additionally, steps such as the creation of the LIMES same-as relations must be done for every relation between two datasets. Next, the system has no capabilities to perform advanced levels of querying. Filtering is not the same is querying. The system cannot perform a query on multiple separate WFS data stores or make multistage filters; for example, it cannot perform the following query: find all objects within 1,000 meters of object X. To complete this query, a system would need to look up object X, return the geographic data for object X in the form of a bounding box, and then search multiple WFS endpoints for objects that are within that bounding box. Additionally, users must have knowledge of the individual contents of the datasets to perform any filters. Lastly, data are difficult to link because of different attributes for each dataset. Currently, this tool only performs entity matching. There is no way to navigate the data between datasets based on different attributes. The heterogeneous nature of the datasets makes it impossible for users to create a filter that

can accurately draw the requested data from different sources because the fields will be different and the possible values could be different.

Conclusions

The objectives of this research and development effort were to investigate technical procedures that can enable semantically supported data integration with The National Map, coreference external data sources with data from The National Map to enrich user experiences, and determine methods of generalization and symbolization that work with a cartographic user interface design supporting LD. The developed prototype system provided data from The National Map that were linked to external data sources, such as DBpedia and Geonames, which provided the integration and linking for further exploration of those datasets and the features of The National Map. The coreferenced data from those external sources and within The National Map sources provide additional information to users about those features. The information in this prototype was delivered by a user interface that accesses LD for its display and visualization function. These visualization functions use simple generalization methods (for example, summing many geographic features to a single symbol for representation at small cartographic scales) but also provide an effective user interface for further access to the characteristics of a feature. This system demonstrates a system design for implementing advanced feature descriptions for a map as a knowledge base.

Challenges and Potential Solutions

The major challenge is the lack of a querying system and the linkability of the datasets. The lack of querying reduces the usability of the tool by reducing the number of ways data can be searched and traversed. Likewise, the lack of indepth linkability between the data elements system wide because of the heterogeneity of the datasets results in reduced usability of the system because users need to know what data are stored and how they are stored for each individual dataset; however, research has already begun in this area. Currently, there are two options to solve this issue. The first option is to mass convert everything into RDF format and perform querying here. RDF format allows increased linkability and already has querying languages in the form of SPARQL. This approach results in increased storage requirements because all the data would need to be stored in RDF format. The second option is to design a querying system for WFS (Zhao and others, 2017); however, this approach is still in its infancy.

Statement of Importance

There are some major pieces of this work that are important. First, this work proposes an easy-to-use platform that can create an LD tool between heterogeneous GIS datasets, which means that it is easy for users to change the datasets, add new datasets, and change how the data are interpreted and compared. Secondly, this work highlights the difficulty of relating datasets even within the same organization; for example, the USGS Structures and USGS GNIS datasets have different standards for describing features. Lastly, this work stands as a template to link USGS datasets with outside datasets such as the Geonames.org data, which allows our already free-use datasets to become closer to being incorporated in the Semantic Web.

Summary

A prototype system to explore Linked Data that semantically integrates geospatial data in various formats from different publication sources with data from The National Map of the USGS is presented. This work is one stage in the development of a map knowledge base from The National Map data. The focus is on accessing advanced feature descriptions for data from The National Map with data coreferenced from other sources. The prototype uses Geoserver to access The National Map data, which are converted to Resource Description Framework triples using Karma and stored in the Marmotta triplestore. Marmotta uses a Postgres relational database as a backend for the project and queries to the Marmotta triplestore are converted to structured query language and executed by Postgres. Once the triples are retrieved, the link discovery framework for metric spaces is used to determine same_as relationships in additional data sources. The system includes linking of data from multiple sources to original data accessed from The National Map. These links provide additional attributes, descriptions, and relationships in popup tables of The National Map features from Web data sources. Data from The National Map used in the prototype include geographic names, structures, and boundaries. Feature from these datasets are then linked to external data sources including to retrieve additional attributes. The original data and results of these queries including linked data and attributes are visualized using Leaflet. Workflows for accessing The National Map data with Geoserver, conversion to triples, query and linking external datasets, and visualization are described in detail allowing users to replicate the system design and processing. The advanced feature description workflow is used to retrieve additional features for any user-specified object.

A use case for the system is provided to access structures and names information from The National Map for the Washington, D.C., area and link these to Geonames data. The case study data are processed through the various workflows and result in a graphical visualization of the original data and results of the linking process including display of additional attribute information.

References Cited

Agile Knowledge Engineering and Semantic Web, 2016, User manual 1.0.0: GitHub web page, accessed July 5, 2019, at http://dice-group.github.io/LIMES/#/user_manual/index.

Agile Knowledge Engineering and Semantic Web, 2018, LIMES—LInk discovery for MEtric Spaces: LIMES web page, accessed August 2, 2019, at http://aksw.org/Projects/LIMES.html.

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

Baumer, W., Powell, L.J., and Varanka, D.E., 2018, Mapping interactive geospatial linked data: St. Louis, Mo., May 14–16, 2018, Free and Open Source Software for Geospatial North American (FOSS4GNA), 4 p.

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

Bulen, A.N., Carter, J.J., and Varanka, D.E., 2011, A program for the conversion of The National Map data from proprietary format to resource description framework (RDF): U.S. Geological Survey Open-File Report 2011–1142, 9 p., accessed November 7, 2019, at https://doi.org/10.3133/ofr20111142.

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/.

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/.

Leaflet, 2019, Leaflet—An open-source JavaScript library for mobile-friendly interactive maps: Leaflet, version 1.5.1, accessed August 2, 2019, at https://leafletjs.com/.

Lewis, R., ed., 2007, Dereferencing HTTP URIs: W3C web page, accessed March 26, 2019, at http://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14.html.

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, 2010, OpenGIS Filter Encoding 2.0 Encoding Standard: OGC web page, accessed November 6, 2019, at https://www.opengeospatial.org/standards/filter.

OGC, 2017, Leaflet—WFST: Leaflet v. 1.1.1 software release, accessed July 8, 2019, at https://github.com/Flexberry/Leaflet-WFST.

OGC, 2019a, Geography markup language: OGC web page, accessed August 2, 2019, at https://www.opengeospatial.org/standards/gml.

OGC, 2019b, GeoPackage: OGC web page, accessed November 5, 2019, at https://www.geopackage.org/.

OGC, 2019c, 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/.

Open Source Geospatial Foundation, 2019c, PostGIS, accessed November 5, 2019, at https://postgis.net/.

Open Source Geospatial Foundation, 2019d, WFS-NG Plugin, accessed November 5, 2019, at https://docs.geotools.org/latest/userguide/library/data/wfs-ng.html.

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.

PostgreSQL Global Development Group, 2019, PostgreSQL—The world's most advanced open source relational database: PostgreSQL database, accessed August 2, 2019, at https://www.postgresql.org/.

Powell, L.J., and Varanka, D.E., 2018, A linked GeoData map for enabling information access: U.S. Geological Survey Open-File Report 2017–1150, 6 p., accessed November 7, 2019, at https://doi.org/10.3133/ofr20171150.

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

Tandy, J., Brink, L. van den, and Barnaghi, P., 2017, Spatial data on the web best practices: W3C web page, accessed November 7, 2019, at https://www.w3.org/TR/sdw-bp/#applicability-formatVbp.

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/.

Usery, E.L., and Varanka, D.E., 2012, Design and development of linked data for The National Map: The Semantic Web Journal, 14 p., accessed November 7, 2019, at http://www.semantic-web-journal.net/content/design-and-development-linked-data-national-map.

U.S. Geological Survey, 2013, The National Map: U.S. Geological Survey digital data, accessed December 18, 2013, at https://nationalmap.gov/. [Also available at https://www.usgs.gov/core-science-systems/national-geospatial-program/national-map.]

U.S. Geological Survey, 2019a, The National Map—Service endpoints: U.S. Geological Survey web page, accessed August 1, 2019, at https://viewer.nationalmap.gov/services/.

U.S. Geological Survey, 2019b, U.S. Board on Geographic Names—Domestic names: U.S. Geological Survey web page, accessed June 27, 2019, at https://www.usgs.gov/core-science-systems/ngp/board-on-geographic-names/domestic-names.

U.S. Geological Survey, 2019c, NGP standards and specifications: U.S. Geological Survey web page, accessed August 1, 2019, at https://www.usgs.gov/core-science-systems/ngp/ss/supporting-themes.

U.S. Geological Survey, 2019d, 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, 2019e, File formats for domestic geographic names: U.S. Geological Survey, 9 p., accessed November 7, 2019, at https://geonames.usgs.gov/docs/pubs/Nat_State_Topic_File_formats.pdf.

U.S. Geological Survey, 2019f, The National Map—USGSTopo (MapServer): U.S. Geological Survey digital data, accessed August 5, 2019, at https://basemap.nationalmap.gov/arcgis/rest/services/USGSTopo/MapServer/.

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

Varanka, D.E., and Usery, E.L., 2018, The map as knowledge base: International Journal of Cartography, v. 4, no. 2, p. 201–223, accessed December 10, 2018, at https://doi.org/10.1080/23729333.2017.1421004.

Vatant, B., and Wick, M., 2012, GeoNames ontology: GeoNames web page, accessed June 28, 2019, at https://www.geonames.org/ontology/documentation.html.

Vretanos, P., ed., 2010, OpenGIS filter encoding 2.0 encoding standard: Open Geospatial Consortium document OGC 09–026r1 and ISO/DIS 19143, 90 p., accessed August 2, 2019, at https://www.opengeospatial.org/standards/filter.

Wick, M., and Boutreaux, C., 2019, GeoNames: GeoNames database, accessed June 27, 2019, at https://www.geonames.org/.

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/.

Zhao, T., Zhang, C., and Li, W., 2017, Adaptive and optimized RDF query interface for distributed WFS data: ISPRS International Journal of Geo-Information, v. 6, no. 4, 108 p. [Also available at https://doi.org/10.3390/ijgi6040108.]

Glossary

advanced feature description Tabular, popup attributes, relations, other text, and links providing details of selected features.

coreference Two or more expressions in a text that refer to the same person or thing; they have the same referent where the referent is a person or thing to which a name—a linguistic expression or other symbol—refers.

data integration Merging geospatial features and attributes based on geometric coordinate reference systems, geospatial position, and semantic descriptions and relation links.

dereference Accessing the value or object in a linked web or memory location stored in a pointer or another value interpreted as such; to access a value being referenced by something else.

For more information about this publication, contact:
Director, USGS National Geospatial Technical Operations Center
1400 Independence Road
Rolla, MO 65401
(573) 308–3500

For additional information, visit: https://www.usgs.gov/core-science-systems/ngp/ngtoc.

Publishing support provided by the Rolla Publishing Service Center

ISSN 2328-0328 (online)
https://doi.org/10.3133/sir20195148