Open Geospatial Consortium

Publication Date: 2015-11-18

Approval Date: 2015-09-15

Posted Date: 2015-07-22

Reference number of this document: OGC 15-054

Reference URL for this document: http://www.opengis.net/doc/PER/t11-linked-data

Category: Public Engineering Report

Editor(s): Stephane Fellah

Testbed-11 Implementing Linked Data and Semantically Enabling OGC Services Engineering Report

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. The ER is distributed for review and comment. This ER is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable

Copyright © 2015 Open Geospatial Consortium.

Contents

Figures

Page

Tables

Page

Abstract

This OGC® Engineering Report (ER) summarizes the approaches, findings and the results of the Linked Data and Semantic Enablement of OGC Web Services sub-thread activities of the OGC Testbed-11 Cross Community Interoperability (CCI) Thread. This report provides an overview of existing standards for geosemantics, outlines the approaches adopted during the testbed, describes the conceptual semantic models and services developed during this testbed to leverage Linked Data and semantic enabled OGC web services.

Business Value

This Engineering Report proposes a solution to improve semantic interoperability in the following areas:

Keywords

ogcdocs, testbed-11, ogcdoc, ogc documents, ows11, ontology, cci, GeoSPARQL, gazetteers, portrayal, symbology, mediation, alignment, semantic, RDF, Linked Data, OWL, SKOS, semantic-enablement.

Testbed-11 Implementing Linked Data and Semantically Enabling OGC Services Engineering Report

1 Introduction

1.1 Scope

The Testbed 11 CCI Thread had multiple sub-threads, in which linked data standards were used to demonstrate semantic interoperability of information model and services. This OGC® Engineering Report summarizes what Semantic Web and Linked data are, outlines the value proposal of using a semantic-based approach as a solution for rapid integration of information and services, outlines best practices defined by the communities using Linked Data. We summarize also the ontologies and semantic based services developed and used during the testbed and we provide a set of recommendations for future testbeds and activities within OGC.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Stephane Fellah	Image Matters LLC
Gobe Hobona	Envitia
Dave Wesloh	NGA

1.3 Future work

For recommendations on future work please refer to section 14.

1.4 Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 11-052r4, *OGC GeoSPARQL –A Geographic Query Language for RDF Data*

OGC 11-063r6, *OWS-8 CCI Semantic Mediation Engineering Report*

OGC 12-103r3, *OWS-9 CCI Semantic Mediation Engineering Report*

OGC 14-029, *OGC® Implementing Linked Data and Semantically Enabling OGC Services Engineering Report*

OGC 14-049, *Testbed 10 OWS CCI Ontology Engineering Report*

OGC 15-057, *Incorporating Social Media in Emergency Response Engineering Report.*

OGC 15-058, *Testbed 11 Symbology Mediation Engineering Report*

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] and in OpenGIS® Abstract Specification shall apply. In addition, the following terms and definitions apply.

3.1

feature

representation of some real world object or phenomenon

3.2

interoperability

capability to communicate, execute programs or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

3.3
metadata
data about data

3.4
model
abstraction of some aspects of a universe of discourse

3.5
ontology
a formal specification of concrete or abstract things, and the relationships among them, in a prescribed domain of knowledge [ISO/IEC 19763]

3.6
semantic interoperability
the aspect of interoperability that assures that the content is understood in the same way in both systems, including by those humans interacting with the systems in a given context

3.7
syntactic interoperability
the aspect of interoperability that assures that there is a technical connection, i.e. that the data can be transferred between systems

4 Conventions

4.1 Abbreviated terms

CCI	Cross Community Interoperability
E&DM	Emergency and Disaster Management
ER	Engineering Report
GML	Geography Markup Language
HTML	HyperText Markup Language
JSON-LD	JavaScript Object Notation for Linked Data
LEAPS	Law Enforcement and Public Safety
NIEM	National Information Exchange Model
OGC	Open Geospatial Consortium
OWL	Web Ontology Language

OWS-8	OGC Web Services Initiative, Phase 8
OWS-9	OGC Web Services Initiative, Phase 9
RDF	Resource Description Framework
SDI	Spatial Data Infrastructure
SE	Symbology Encoding
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
SVG	Scalable Vector Graphics
URI	Unique Resource Identifier
URL	Uniform Resource Locator
URL	Uniform Resource Name
WFS	Web Feature Service
WKT	Well Known Text
WMS	Web Map Service

4.2 UML notation

Some diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

5 ER Topic overview

This OGC Testbed 11 Engineering Report (ER) addresses the use of Linked Data standards to represent, share geospatial information and semantically enabling OGC services. The lack of semantics and semantic-based services remains one of the greatest impediments to cross-community interoperability (CCI) between heterogeneous information models and service protocols. Many CCI efforts today start by defining and standardizing logical data models, which in turn drive the implementation of dictionaries (how to call things), metadata (a means to describe and discover things) and services (how to access things). Such “data-centric” approaches restrict standardization efforts to mostly syntactic and schematic considerations, towards the goal of establishing common schemata and protocols for interoperability. These approaches are limited by their syntactic and schematic boundaries, and fail to provide sufficient semantics (explicit meaning of concepts) and context (relevant associated information) for machines, which are key to enabling enhanced interoperability and richer information exchange. What's needed is an approach where semantics and context contribute to a rich, conceptual model that formally encodes logical data model semantics and context as ontologies, building upon heterogeneous logical data model foundations. This new "semantic layer" conveys how geospatial entities should be properly interpreted and employed in a business context, thus contributing to a higher level of utility, interoperability and automation.

The Testbed 11 CCI Thread had multiple sub-threads, in which linked data standards were used to demonstrate semantic interoperability of information model, services. This engineering report summarizes what Semantic Web and Linked data are, outlines the value proposal of using a semantic-based approach as a solution for rapid integration of information and services, outlines best practices defined by the communities. We also summarize the ontologies and semantic based services developed and used during the testbed and we provide a set of recommendations for future testbeds and activities within OGC.

6 Overview of Semantic Web

6.1 Semantic Web

W3C defines the *Semantic Web is a Web of data*. There is a lot of data we all use every day, and it's not part of the Web. For example, I can see my bank statements on the web, and my photographs, and I can see my appointments in a calendar. But can I see my photos in a calendar to see what I was doing when I took them? Can I see bank statement lines in a calendar? Why not?

- Because we don't have a web of linked data.
- Because data is controlled by applications, and each application keeps data to itself.

The vision of the Semantic Web is to extend principles of the Web from documents to data. Data should be accessed using the general Web architecture e.g., using URI-s: data should be related to one another just as documents (or portions of documents) are already. This also means creation of a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. Data should be processed automatically by tools as well as manually, including revealing possible new relationships among pieces of data.

Semantic Web technologies can be used in a variety of application areas. For example:

- In *data integration*, whereby data in various locations and various formats can be integrated in one, seamless application;

- In *resource discovery and classification* to provide better, domain specific search engine capabilities;
- In *cataloging* for describing the content and content relationships available at a particular Web site, page, or digital library;
- By *intelligent software agents* to facilitate knowledge sharing and exchange;
- In *content rating*; in describing *collections* of pages that represent a single logical "document";
- For describing *intellectual property rights* of Web pages (see, eg, the Creative Commons), and in many others.

The list of Semantic Web Case Studies and Use Cases[1] gives some further examples.

6.2 Why Semantics?

Table 1 summarizes the value proposition of using semantic web technologies to address the challenges of interoperability of information and services.

Table 1 Value Proposition of Semantic Web

Issue in Data Centric Approach	How Linked Data Centric addresses it	Value Proposition
Data Schema Standardization of domain model uses a syntactic approach tending to minimize heterogeneity, imposes strict (unforgiving) adherence to standard to interoperate. It is a painful and long process due to the need of building consensus.	The semantic-based approach embraces the heterogeneity of domain models by providing a common formal, sharable framework mechanism to easily extend meta model for accommodating specific needs. The extensions can be done in a decentralized way without breaking the existing infrastructure.	• Decentralized extension of the model. • Accommodation of model specificities • Shareable and machine processable model and business rules
Data Schemas have limited expressiveness. **Data Schema captures only the syntactic and structural constraints of data model, but does not provide machine-processable conceptual model and business rules. Implementers are required to hardcode the rules with the risk of having different interpretations of a written specification of the rules.**	OWL and SPARQL Rules are providing a standard-based mechanism to capture formal conceptual models along with their business rules in machine-processable way, meaning that could be imported by system implementation without writing code.	• Reduction of software development cost • Exchangeable machine processable rules and conceptual models, which allows automation and reduction of code. • Unambiguous interpretation of domain model

---

[1] docs.opengeospatial.org/dp/15-074/15-074.html

[2] The act of retrieving a representation of a resource identified by a *URI* is known as *dereferencing* that *URI*. (W3C

Evolution of domain model and associated software is difficult when using data-centric approach due to the fact that business rules and semantic of data model are hardcoded in application. Any new changes in standard require expensive update of software. Very often evolution of data model requires building consensus and standardization, which can be a very lengthy process.	Ontology provides a framework to extend metamodel in a decentralized way and accommodate the specificity of each domain players. The extensions can be integrated and handled by any generic-purpose semantic-based reasoners and validators without rewriting code. Software can adapt quickly to model and business rules changes.	☐ Cost reduction in software updates ☐ Software adapts and evolves to domain model changes without rewriting code. ☐ Competitiveness gain in fast changing environment ☐ Decentralized and organic evolution of domain model
Integration and interoperability with other domains is difficult due to the heterogeneity of data schemas and business models, the lack of common protocols and machine-processable conceptual model and business rules.	RDF, RDFS/OWL and SPARQL standards provide a universal mechanism to exchange knowledge, schemas/rules and query knowledge over the web. Knowledge representation can be layered on top of existing information assets, which means they are an ***enhancement*** and not a displacement for prior investments. Due to the inherent connectedness of the RDF model, different domain models can be linked together into creative mashups to create added-value.	☐ Knowledge-based approach is accommodating and enhancing existing systems ☐ Open standards (RDF,RDFS/OWL,SPARQL) ☐ Web Scalability ☐ Low-level cost for mashup with different domains (Infrastructure, Counter-Terrorism, HADR).
Inferencing and validation on data is limited and often hardwired in software code. The inferences and validation rules are typically captured in a non-processable way (documentation, UML diagrams, schema annotations) which is let to the interpretation of the readers.	Because OWL provides a standard formal metamodel with well-defined semantics, off-the shelf reasoners and rules engines can be used to automate the inferencing and validation of data.	☐ Automatic inferencing and validation can be done using off-the-shelf reasoners and rule engines. ☐ Reduction of validation and testing lifecycle.
Code duplication: Each application must express the semantics and business rules by burying it into code.	By making the semantic and business rules of domain model formalized unambiguously into a declarative ontology, software don't need to hardcode the rules and interpretation of the data into the code, but instead can import the ontologies to properly get the inferences from the data using off-the-self semantic based components.	☐ Reduction of complexity of software code ☐ Easy maintenance of software ☐ Cost reduction in software development.
Variability of interpretation: Because there are no machine-processable business rules, each application can take its own interpretation. For example: Mars probe disaster - one application interpreted the data in inches, another application interpreted the data in centimeters.	By making the semantic and business rules of domain model formalized unambiguously into a declarative ontology, software can use off-the shelf semantic components to import the ontologies and rules to interpret, infer and validate knowledge-based representation of the domain and thus reduce the cost of errors of interpretation.	☐ Robust software ☐ Reduction in maintenance cost. ☐ Machine processable business rules
No ad-hoc discovery and exploitation: Applications have the semantics pre-wired. Thus, when new data is encountered an application may not be able to effectively process it. This makes for brittle applications.	By making conceptual model and rules explicitly represented using standard ontology (OWL), software can remove hardwire semantic and use off-the-shelf semantic component to process effectively changes in model without rewriting codes making it more robust to change.	☐ Cost reduction in software maintenance ☐ Software adapts and evolves to domain model changes without rewriting code.

6.3 Semantic Web Stack

The Semantic Web Stack is an illustration of the hierarchy of languages, where each layer exploits and uses capabilities of the layers below. It shows how technologies that are standardized for the Semantic Web are organized to make the Semantic Web possible. The stack also shows how the Semantic Web is an extension (not replacement) of classical hypertext web.

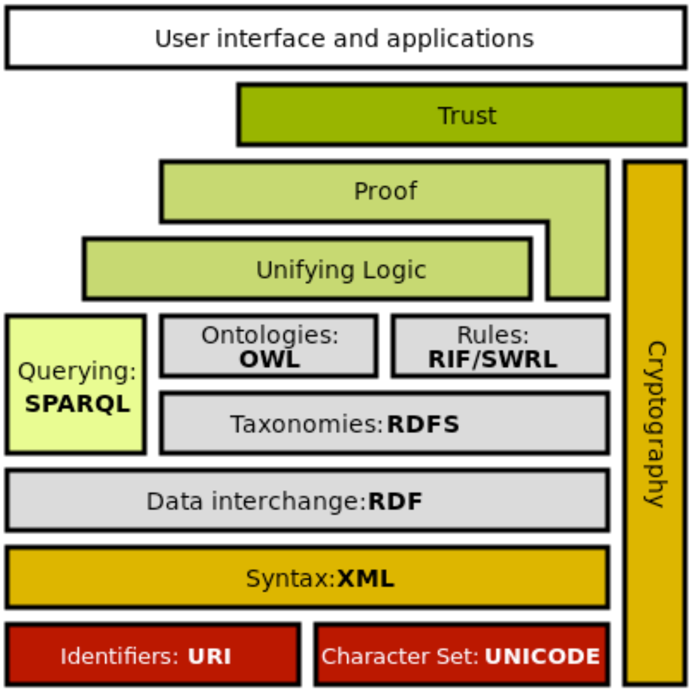


Figure 1 Semantic Web Stack

The Semantic Web Stack illustration in Figure 1 was created by Tim Berners-Lee. The stack is still evolving as the layers become more formalized.

As shown in the Semantic Web Stack, the following languages or technologies are used to create the Semantic Web. The technologies from the bottom of the stack up to OWL are currently standardized and accepted as tools and technologies to build Semantic Web applications. Currently, however, how the top of the stack is going to be implemented is not clear. All layers of the stack need to be implemented to achieve the full vision of the Semantic Web.

The bottom layers contain technologies that are well known elements of the hypertext web and that without change provide the foundation for the semantic web.

- **Internationalized Resource Identifier (IRI)**: Generalization of URIs provides means for uniquely identifying semantic web resources. The Semantic Web needs unique identification functionality to allow provable manipulation with resources in the top layers.

- **Unicode** serves to represent and manipulate text in many languages. The Semantic Web should also help to bridge documents in different human languages. Therefore the Semantic Web should be able to represent these languages and related transformations..

- **XML** is a markup language that enables creation of documents composed of structured data. The Semantic web gives meaning (semantics) to structured data.

- **XML Namespaces** provides a way to use markups from more sources. The Semantic Web is about connecting data together. Therefore more sources in one document need to be able to be referenced.

Middle layers in the stack contain technologies standardized by the W3C to enable building semantic web applications.

- **Resource Description Framework (RDF)** is a framework for creating statements in a form of so-called triples. This enables the representation of information about resources in the form of graph - the semantic web is sometimes called Giant Global Graph.

- **RDF Schema (RDFS)** provides basic vocabulary for RDF. Using RDFS it is possible to create hierarchies of classes and properties.

- **Web Ontology Language (OWL)** extends RDFS by adding more advanced constructs to describe semantics of RDF statements. OWL allows stating additional constraints, such as cardinality, restrictions of values, or characteristics of properties such as transitivity. OWL is based on description logic and so brings reasoning power to the semantic web.

- **SPARQL** is a RDF query language. SPARQL can be used to query any RDF-based data (i.e., including statements involving RDFS and OWL). Querying language is necessary to retrieve information for semantic web applications.

- **RIF** is a rule interchange format. RIF is important, for example, because it allows describing relations that cannot be directly described using description logic such as used in OWL.

The top layers in the stack contain technologies that are not yet standardized or contain just ideas that should be implemented in order to realize the full Semantic Web.

- **Cryptography** is important to ensure and verify that semantic web statements are coming from trusted sources. This can be achieved by appropriate digital signaturea of RDF statements.
- **Trust** in derived statements will be supported by (a) verifying that the premises came from trusted source and by (b) relying on formal logic during deriving new information.
- **User interface** is the final layer that will enable humans to easily use semantic web applications.

7 Linked Data

7.1 What is Linked Data?

The Web enables us to link related documents. Similarly it enables us to link related data. The idea of Linked Data is more recent (2006) than the Semantic Web (2001), but sometimes it is easier to think of the semantic web as building on the ideas behind Linked Data. The term **Linked Data** refers to a set of best practices for publishing and connecting structured data on the Web. Key technologies that support Linked Data are URIs (a generic means to identify entities or concepts in the world), HTTP (a simple yet universal mechanism for retrieving resources, or descriptions of resources), and RDF (a generic graph-based data model with which to structure and link data that describes things in the world). There are not specific standards or specifications for Linked data. Instead, Linked Data is defined by a **set of best practices** for providing a data infrastructure that makes it easier to share data across the web. Linked Data APIs provide a common access mechanism for data on the Web which is more convenient than many separately and differently designed APIs published by individual data suppliers. You can then use semantic web technologies such as RDFS, OWL and SPARQL to build application around data and the LD API.

7.2 Four principles of linked data

Tim Berners-Lee, the inventor of the Web and initiator of the Linked Data project, outlined **four principles for linked data** in his Design Issues: Linked Data note published in 2006 (his wording are bolded and we added our own commentaries)

- **Use URIs to name things**: URIs are the best available way to uniquely identify things and therefore to identify connections between things. If it is not URIs, it is not a semantic web.
- **Use HTTP URIs so that things can be referred to and looked up ("dereferenced") by people and user agents**. You may have seen URIs that begin with different schemas such as ftp:, mailto, :urn, or a prefix made up a

particular community. Using these other approaches reduces interoperability, and interoperability is what it’s all about.

- **When someone looks up a URI, provide useful information, using the open Web standards such as RDF, RDFS, OWL, SPARQL**. A URI can just be a name and not actually the address of a web page. This principle says that you may as well put something there, such as an HTML page, or something else. Whatever is provided should use a recognized standard. RDFS and OWL let you define a vocabulary of terms and information about those terms and their relationships in a machine-readable way (for example by SPARQL queries). Because of this, if a URI that identifies a resource leads to RDFS or OWL declarations about that resource, this is a big help to applications. Large datasets should provide a SPARQL query service, but the basic linked data should be provided as well.

- **Include links to other related things using their URIs when publishing on the Web.** In hypertext web sites it is considered generally rather bad etiquette not to link to related external material. The value of your own information is very much a function of what it links to, as well as the inherent value of the information within the web page. in the same is true for the Semantic Web. In addition to HTML linking element, various RDF vocabularies provide other properties as a way to say “this data (or this element of data) has a specific relationship to another resource on the web”. When applications can follow these links, they can do interesting new things.

7.3 5-Star Linked Data

In a talk at the 2010 Gov 2.0 Expo in Washington,D.C., Berners-Lee gave a fairly nontechnical introduction to Linked Data in which he suggested the awarding of stars to governments for sharing data on the web. The **5 Star Linked Open Data** refers to an incremental framework for deploying data. The 5 Star Linked Data system is cumulative. Each additional star presumes the data meets the criteria of the previous step(s). 5 Star Linked Open Data includes an **Open License** (expression of rights) and assumes publications on the public Web.

Organizations may elect to publish **5 Star Linked Data**, without the word "open", implying that the data **does not** include an Open License (expression of rights) and does not imply publication on the public Web.

☆ **Publish data on the Web in any format (e.g., PDF, JPEG) accompanied by an explicit Open License (expression of rights).**

☆☆ **Publish structured data on the Web in a machine-readable format (e.g., XML).**

☆☆☆ **Publish structured data on the Web in a documented, non-proprietary data format (e.g., CSV, KML).**

☆☆☆☆ **Publish structured data on the Web as RDF (eg Turtle, RDFa, JSON-LD, SPARQL)**

☆☆☆☆☆ **In your RDF, have the identifiers be links (URLs) to useful data sources**.

7.4 Linked Data Best Practices

Over the years, a number of best practices have emerged to publish and consume linked data information on the web. This section summarizes some of the best practices that have been published in several documents published by W3C.

7.4.1 Best Practices for Publishing Linked Data

The following best practices are discussed in this document and listed here for convenience.

7.4.1.1 Good URIs for Linked Data

The core of Linked Data is a well-considered URI naming strategy and implementation plan, based on HTTP URIs. Consideration for naming objects, multilingual support, data change over time and a persistence strategy are the building blocks for useful Linked Data.

Resources are named with URI references. When publishing Linked Data, you should devote some effort to choosing good URIs for your resources.

On the one hand, they should be *good names* that other publishers can use confidently to link to your resources in their own data. On the other hand, you will have to put technical infrastructure in place to make them ***dereferenceable***[2], and this may put some constraints on what you can do.

This section lists, in loose order, some things to keep in mind.

- Use HTTP URIs for everything. The http:// scheme is the only URI scheme that is widely supported in today's tools and infrastructure. All other schemes require extra effort for resolver web services, dealing with identifier registrars, and so on. The arguments in favor of using HTTP are discussed in several places, e.g. in Names and addresses by Norman Walsh, and URNs, Namespaces and Registries (draft) by the W3C Technical Architecture Group (TAG).

---

[2] The act of retrieving a representation of a resource identified by a *URI* is known as *dereferencing* that *URI*. (W3C 2007)

- Define your URIs in an HTTP namespace under your control, where you actually can make them dereferenceable. Do not define them in someone else's namespace.

- Keep implementation cruft out of your URIs. Short, mnemonic names are better. Consider these two examples:

http://dbpedia.org/resource/Berlin

http://www4.wiwiss.fu-berlin.de:2020/demos/dbpedia/cgi-bin/resources.php?id=Berlin

- Try to keep your URIs stable and persistent. Changing your URIs later will break any already-established links, so it is advisable to devote some extra thought to them at an early stage.

- The URIs you choose are constrained by your technical environment. If your server is called demo.serverpool.wiwiss.example.org and getting another domain name is not an option, then your URIs will have to begin with http://demo.serverpool.wiwiss.example.org/. If you cannot run your server on port 80, then your URIs may have to begin with http://demo.serverpool.example.org:2020/. If possible you should clean up those URIs by adding some URI rewriting rules to the configuration of your webserver.

We often end up with three URIs related to a single non-information resource:

- an identifier for the resource,

- an identifier for a related information resource suitable to HTML browsers (with a web page representation),

- an identifier for a related information resource suitable to RDF browsers (with an RDF/XML representation).

Here are several ideas for choosing these related URIs:

http://dbpedia.org/resource/Berlin

http://dbpedia.org/page/Berlin

http://dbpedia.org/data/Berlin

Or:

http://id.dbpedia.org/Berlin

http://pages.dbpedia.org/Berlin

http://data.dbpedia.org/Berlin

Or:

http://dbpedia.org/Berlin

http://dbpedia.org/Berlin.html

http://dbpedia.org/Berlin.rdf

You will often need to use some kind of primary key inside your URIs to make sure that each one is unique. If you can, use a key that is meaningful inside your domain. For example, when dealing with books, making the ISBN number part of the URI is better than using the primary key of an internal database table. This also makes equivalence mining to derive RDF links easier.

Examples of cool URIs:

http://dbpedia.org/resource/Boston

http://www4.wiwiss.fu-berlin.de/bookmashup/books/006251587X

7.4.1.2 Use standard Vocabularies

Whenever possible, describe objects with previously defined vocabularies. Extend standard vocabularies where necessary, and create vocabularies (only when required) that follow best practices whenever possible. When you cannot find good existing vocabularies that cover all the classes and properties you need, then you have to define your own terms. Defining new terms is not hard. RDF classes and properties are resources themselves, identified by URIs, and published on the Web, so everything we said about publishing Linked Data applies to them as well. You can define vocabularies using the RDF Vocabulary Description Language 1.0: RDF Schema or the Web Ontology Language (OWL).

Here we give some guidelines for those who are familiar with these languages:

- **Do not define new vocabularies from scratch,** but complement existing vocabularies with additional terms (in your own namespace) to represent your data as required.

- **Provide for both humans and machines.** At this stage in the development of the Web of Data, more people will be coming across your code than machines, even though the Web of Data is meant for machines in the first instance. Don't forget to add prose, e.g. rdfs:comments for each term invented. Always provide a label for each term using the rdfs:label property.

- **Make term URIs dereferenceable.** It is essential that term URIs are dereferenceable so that clients can look up the definition of a term. Therefore you should make term URIs dereferenceable following the W3C Best Practice Recipes for Publishing RDF Vocabularies.

- **Make use of other people's terms**. Using other people's terms, or providing mappings to them, helps to promote the level of data interchange on the Web of Data, in the same way that hypertext links built the traditional document Web. Common properties for providing such mappings are rdfs:subClassOf orrdfs:subPropertyOf.

- **State all important information explicitly.** For example, state all ranges and domains explicitly. Remember: humans can often do guesswork, but machines can't. Don't leave important information out!

- **Do not create over-constrained, brittle models; leave some flexibility for growth.** For instance, if you use full-featured OWL to define your vocabulary, you might state things that lead to unintended consequences and inconsistencies when somebody else references your term in a different vocabulary definition. Therefore, unless you know exactly what you are doing, use RDF-Schema to define vocabularies.

In recent years, governments worldwide have mandated publication of open government content to the public Web for the purpose of facilitating open societies and to support governmental accountability and transparency initiatives. In order to realize the goals of open government initiatives, the W3C Government Linked Data Working Group has published *Best Practices for Publishing Linked Data*{8] to aid in the access and re-use of open government data. Linked Data provides a simple mechanism for combining data from multiple sources across the Web. Linked Data addresses many objectives of open government transparency initiatives through the use international Web standards for the publication, dissemination and reuse of structured data. We strongly recommend the reader to consult this document to adopt these best practices when they want to published Linked Data.

The World Wide Web Consortium (W3C) has also recently published the first public working draft of its **Data on the Web Best Practices []**. These best practices aims to facilitate interaction between publishers and consumers by providing guidance to publishers that will improve consistency in the way data is managed. Also, promoting the re-use of data and fostering trust in the data among developers, whatever technology they choose to use, will enable the continued expansion of the Web as a medium for the exchange of data.

The openness and flexibility of the Web creates new challenges for data publishers and data consumers. In contrast to conventional databases, for example, where there is a single data model to represent the data and a database management system (DBMS) to control data access, data on the Web allows for the existence of multiple ways to represent and to access data. Furthermore, publishers and consumers may be unknown to each other and be part of entirely disparate communities with different norms and in-built assumptions so that it becomes essential to provide information about data structure, quality, provenance and any terms of use.

The following diagram summarizes some of the main challenges faced when publishing or consuming data on the Web. These challenges were identified from the W3C DWBP Use Cases and Requirements [5] and are described by one or more questions. As presented in the diagram, each one of these challenges is addressed by one or more best practices in the document.

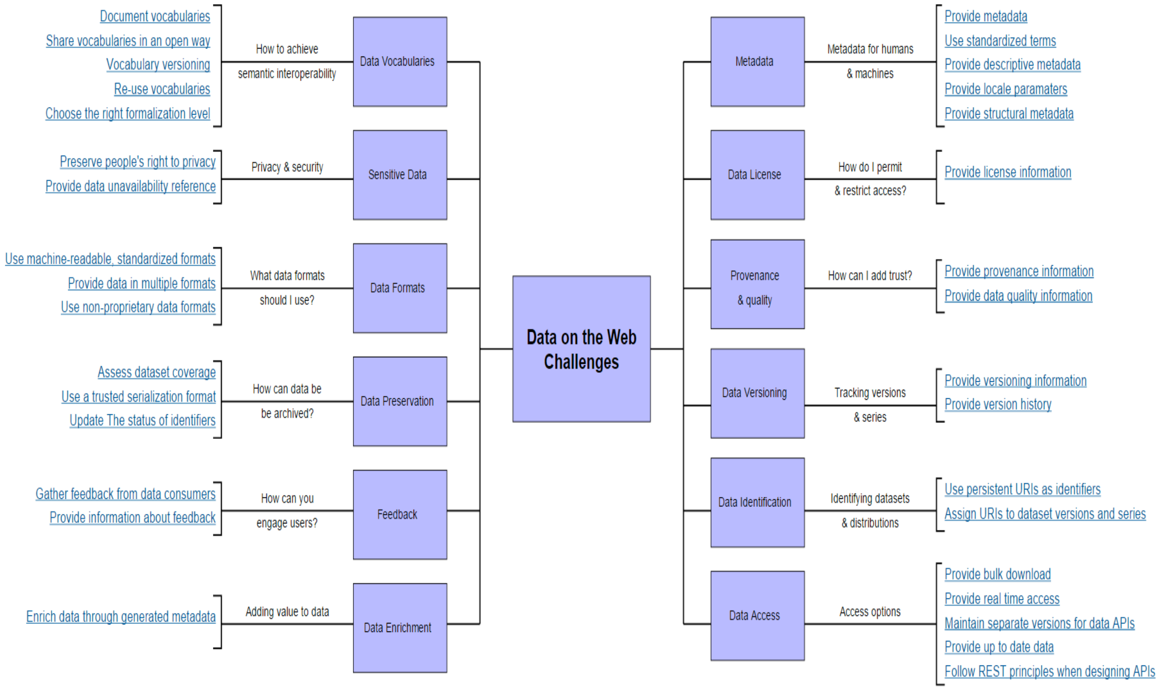


Figure 2 Data on the Web Challenges

These best practices are to be used by data publishers in order to help them and data consumers to overcome the different challenges faced when publishing and consuming data on the Web. One or more best practices were proposed for each one of the previously described challenges. Each BP is related to one or more requirements from the Data on the Web Best Practices Use Cases & Requirements document [5].

8 Review of relevant standards and best practices for Semantic Enablement

8.1 Linked Data Platform API

Linked Data Platform (LDP) is a Linked Data specification defining a set of integration patterns for building RESTful HTTP services that are capable of read-write of RDF data. On 26 February 2015, the W3C Linked Data Platform 1.0 was approved as a W3C Recommendation. The Linked Data Platform allows use of RESTful HTTP to consume, create, update and delete both RDF and non-RDF resources. In addition, it defines a set of "Container" constructs—buckets into which documents can be added with a relationship between the bucket and the object similar to the relationship between a feature collection and its constituent features.

Because the Linked Data Platform (LDP) builds upon the classic HTTP request and response model, and because it aligns well with things like REST, Ajax, and JSON-LD, mainstream web developers may soon find it much easier to leverage the power and benefits of Linked Data. It is too early to know how big of an impact it will actually make, but we anticipate that LDP is going to be an important bridge across the ever-shrinking gap between todays Web of hyperlinked documents and the emerging Semantic Web of Linked Data.

8.2 SPARQL

SPARQL Protocol and RDF Query Language (SPARQL) defines a query language for RDF data, analogous to the Structured Query Language (SQL) for relational databases. It is a family of standards of the World Wide Web Consortium. On 15 January 2008, SPARQL 1.0 became an official W3C Recommendation, and SPARQL 1.1 in March, 2013. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

8.3 OGC GeoSPARQL

OGC GeoSPARQL defines filter functions for geographic information system (GIS) queries using well-understood OGC standards (GML, WKT, etc.).

9 Review of relevant ontologies

9.1 VoID

The Vocabulary of Interlinked Datasets (VoID) is concerned with *metadata about RDF datasets*. It is an RDF Schema vocabulary that provides terms and patterns for describing RDF datasets and is intended as a bridge between the publishers and users of RDF data. VoiD descriptions can be used in many situations, ranging from data discovery to cataloging and archiving of datasets, but most importantly it helps users find the right data for their tasks.

VoiD covers four areas of metadata:

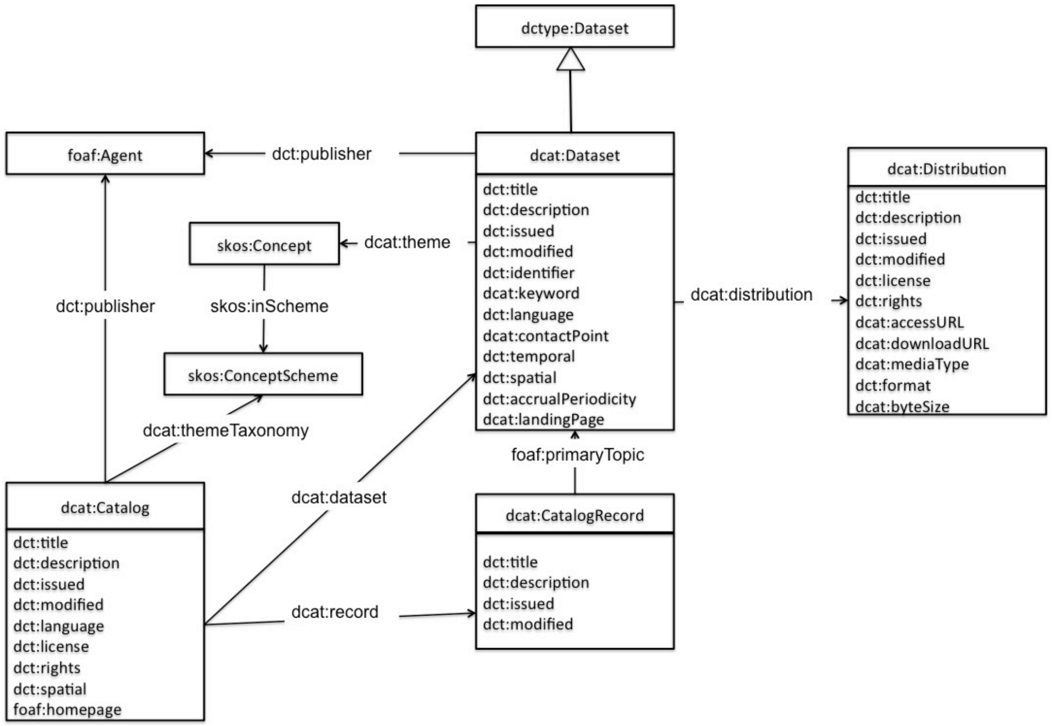
- **General metadata** following the Dublin Core model.
- **Access metadata** describes how RDF data can be accessed using various protocols.

- **Structural metadata** describes the structure and schema of datasets and is useful for tasks such as querying and data integration.

- **Description of links between datasets** are helpful for understanding how multiple datasets are related and can be used together.

9.2 DCAT

DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. By using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume metadata from multiple catalogs. It further enables decentralized publishing of catalogs and facilitates federated dataset search across sites. Aggregated DCAT metadata can serve as a manifest file to facilitate digital preservation.



The DCAT Application profile for data portals in Europe (DCAT-AP) is a specification based on the Data Catalogue vocabulary (DCAT) for describing public sector datasets in Europe. Its basic use case is to enable cross-data portal search for data sets and make public sector data better searchable across borders and sectors. This can be achieved by the exchange of descriptions of datasets among data portals. The application profile is a specification for metadata records to meet the specific application needs of data portals in Europe while providing semantic interoperability with other applications on the basis of reuse of established controlled vocabularies (e.g. EuroVoc) and mappings to existing metadata vocabularies (e.g. Dublin Core, SDMX, INSPIRE metadata, etc).

9.3 SHACL

SHACL (Shapes Constraint Language) is an RDF vocabulary for describing RDF graph structures. These graph structures are captured as "shapes", which are expected to correspond to nodes in RDF graphs. These shapes identify predicates and their associated cardinalities and datatypes. Additional constraints can be associated with shapes using SPARQL or other languages which complement SHACL. SHACL shapes can be used to communicate data structures associated with some process or interface, generate or validate data, or drive user interfaces.

Most applications that share data do so using prescribed data structures. While RDFS and OWL enable one to make logical assertions about the objects in some domain, SHACL (Shapes Constraint Language) describes data structures. Features of SHACL include:

An RDF vocabulary to define shapes as structural declarations of the property constraints associated with those shapes. Complex constraints can be expressed in extension languages like SPARQL.SHACL is based on RDF and is compatible with Linked Data principles, making it possible to mix SHACL shapes with other semantic web data. SHACL definitions are represented in RDF and can be serialized in multiple RDF formats.

9.4 SKOS

SKOS—Simple Knowledge Organization System—provides a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other similar types of controlled vocabulary. As an application of the Resource Description Framework (RDF), SKOS allows concepts to be composed and published on the World Wide Web, linked with data on the Web and integrated into other concept schemes.

In basic SKOS, conceptual resources (concepts) are identified with URIs, labeled with strings in one or more natural languages, documented with various types of note, semantically related to each other in informal hierarchies and association networks, and aggregated into concept schemes.

In advanced SKOS, conceptual resources can be mapped across concept schemes and grouped into labeled or ordered collections. Relationships can be specified between concept labels. Finally, the SKOS vocabulary itself can be extended to suit the needs of particular communities of practice or combined with other modeling vocabularies.

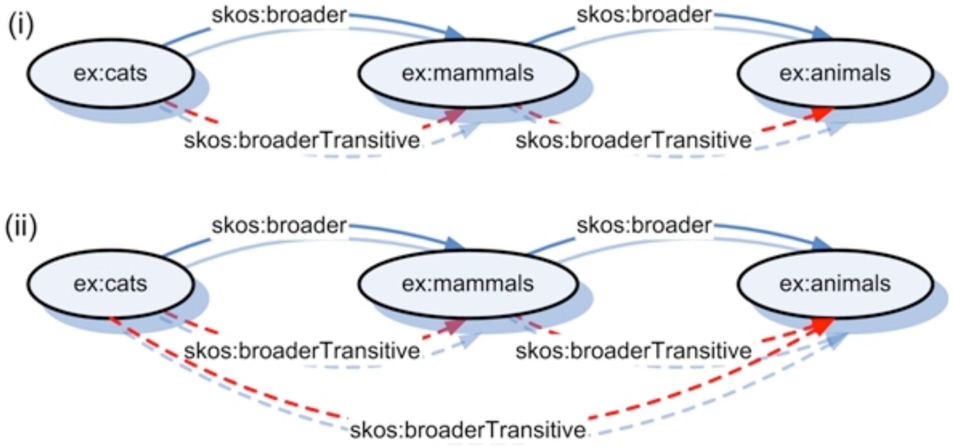


Figure 3 SKOS semantic relationships

Dotted arrows represent statements inferred from the SKOS data model. Solid arrows represent asserted statements.

9.5 SPARQL Service Description

The SPARQL 1.1 Service Description specification provides a vocabulary for describing SPARQL services capabilities that could be used for discovery of SPARQL services and datasets. It provides mechanism to discover the supported query languages (SPARQL 1.0, SPARQL 1.1, SPARQL Update), function extensions, entailment regime (inference) and details about the available datasets managed by the SPARQL endpoint. The SPARQL Service Description is accessible through HTTP Get and can returned in RDF encoding (Turtle, N3, RDF/XML,JSON-LD).

To access the SPARQL Service description, SPARQL services made available via the SPARQL Protocol should return a service description document at the service endpoint when dereferenced using the HTTP GET operation without any query parameter strings provided. This service description must be made available in an RDF serialization (Turtle, RDF/XML, N3,JSON-LD), may be embedded in (X)HTML by way of RDFa, and should use content negotiation if available in other RDF representations.

9.6 RDF Cube

Statistical data is a foundation for policy prediction, planning and adjustments and underpins many of the mash-ups and visualizations we see on the web. There is strong interest in being able to publish statistical data in a web-friendly format to enable it to be linked and combined with related information.

At the heart of a statistical dataset is a set of observed values organized along a group of dimensions, together with associated metadata. The Data Cube vocabulary enables such information to be represented using the W3C RDF (Resource Description Framework) standard and published following the principles of linked data. The vocabulary is based upon the approach used by the SDMX ISO standard for statistical data exchange. This *cube* model is very general and so the Data Cube vocabulary can be used for other data sets such as survey data, spreadsheets and OLAP data cubes [OLAP].

The Data Cube vocabulary is focused purely on the publication of multi-dimensional data on the web.

9.7 PROV

Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness. The PROV Family of Documents defines a model (see Figure 4), corresponding serializations and other supporting definitions to enable the inter-operable interchange of provenance information in heterogeneous environments such as the Web.

In PROV, physical, digital, conceptual, or other kinds of thing are called *entities*. Examples of such entities are a web page, a chart, and a spellchecker. Provenance records can describe the provenance of entities, and an entity's provenance may refer to many other entities.

Activities are how entities come into existence and how their attributes change to become new entities, often making use of previously existing entities to achieve this. They are dynamic aspects of the world, such as actions, processes, etc. For example, if the second version of document D was generated by a translation from the first version of the document in another language, then this translation is an activity.

An *agent* takes a role in an activity such that the agent can be assigned some degree of responsibility for the activity taking place. An agent can be a person, a piece of software, an inanimate object, an organization, or other entities that may be ascribed responsibility. When an agent has some responsibility for an activity, PROV says the agent was *associated* with the activity, where several agents may be associated with an activity and vice-versa.

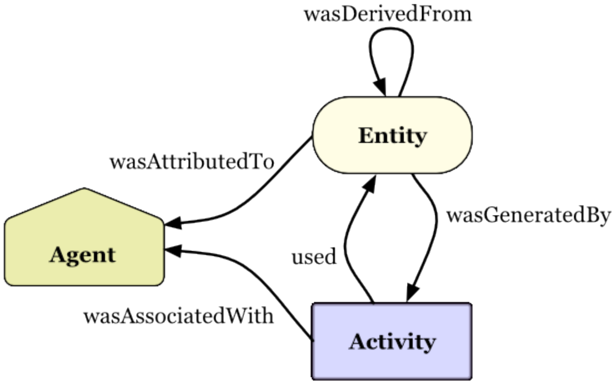


Figure 4 Core PROV Ontology

Provenance of features at dataset, feature and attributes level was investigated during the Testbed 10. Further investigation needs to be done in future testbed.

10 Proposed ontologies

10.1 Core Geospatial Ontologies

The OGC and ISO TC/211 have done tremendous work in modeling geographic information by publishing abstract models (ISO 191xx) and implementation standards for these models (GML, KML, etc.), as well as application schemas (e.g. CityGML). These geospatial data standards serve an important purpose but are inadequate when attempting to automate interconnection of geospatial information because they do not completely tackle the semantic interoperability layer. To address this standards gap, ad-hoc working groups outside of OGC have created ontologies for geospatial information such as GeoJSON and GeoRSS, Ordnance Survey spatial relationships, Geonames ontology and taxonomies for feature types. Unfortunately, these ontologies are neither comprehensive nor cohesive. Existing geospatial standards bodies such as the OGC and ISO TC/211 are just beginning to define abstract geospatial models using taxonomy languages (SKOS) and ontology languages such as RDF/S or OWL, and standardizing the query language for the Geospatial Semantic Web, GeoSPARQL. GeoSPARQL fails to provide a comprehensive set of ontologies to describe geospatial data. Existing geospatial data standards such as GML focus on the structure and encoding syntax of geospatial data, enabling syntactic interoperability of geospatial data, but these geospatial data standards fail to capture model semantics that are crucial to linking to semantic content from other domains.

A comprehensive unified and extensible semantic framework is needed to represent data and metadata that enables true cross-domain interoperability, provides a scalable architecture to integrate heterogeneous geospatial data, and links datasets from other domains (healthcare, law enforcement, public safety, etc.). Web standards (URI, HTTP) and Semantic Web standards (RDF, RDFS, OWL, SKOS, SPARQL, RIF, Linked Data API, etc.) provide the necessary foundation to enable this level of system interoperability. NGA can leverage these standards and improve upon a set of core ontologies for geospatial information, upon which Community Of Interests (COIs) can build and extend to facilitate the semantic interoperability of cross-domain heterogeneous geospatial information.

Image Matters has identified, designed and formalized a set of modular geospatial core and cross-domain ontologies in OWL version 2. The ontologies, illustrated in Figure 1, include mereotopology[3], spatial relations, locations, features, temporal ontologies, geometries, CRS, events, and measures. These "ontology components" provide a core ontological foundation for geospatial information that is universally applicable to any domain. These core ontologies leverage existing standard abstract models (ISO 19xxx), but are modularized and adapted to better leverage the expressiveness of OWL and favor reusability. The resulting Geospatial Ontology can be used as a ontological foundation for all common geospatial information that could be used across various domains (e.g. Defense, Intelligence, Gazetteer , E&DM, Law Enforcement, Public Safety, Hydrology, Aviation).

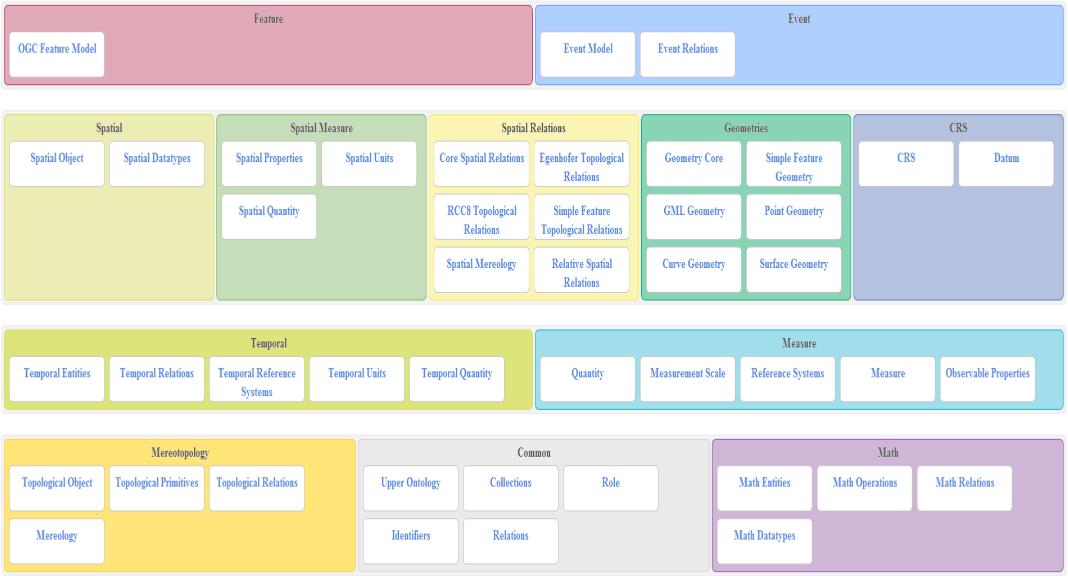


Figure 5 Overview of the Geospatial Ontologies

---

[3] In formal ontology, a branch of metaphysics, and in ontological computer science, **mereotopology** is a first-order theory, embodying mereological and topological concepts, of the relations among wholes, parts, parts of parts, and the boundaries between parts.

More information about the ontology can be found in the OWS-10 Ontology Engineering Report (OGC 14-049). The ontologies are made accessible online at http://ows.usersmarts.com/geospatial.

10.2 SocialML Ontologies

The rapid emergence of Web 2.0 social sites quickly led to an array of new social objects, activities and personas that could better accommodate a broader scoped social experience. This social experience surpassed simple tagging operations found in early social bookmarking sites (delici.o.us) and led Image Matters to the realization that a common model was needed across various types of social media to capture and integrate social information, to build a richer Persona description of an individual or community. This persona includes profile information, cognitive characteristics (skills, beliefs, expertise, interests, goals), relationships, interactions with other members of communities and social objects, rich descriptions of social objects, and influences and roles in different communities. Image Matters, under a DARPA contract, created a comprehensive, extensible and semantic framework for Social Media and networks (SocialML). This framework may be used to model Social Media information. SocialML also serves as the basis for performing semantic-based geo-social analytics and overcoming interoperability barriers between Social Media sites.

Absent a common model to describe the heterogeneity of social media, social web sites remain isolated silos limited by their own APIs and data representations (e.g., Google OpenSocial, Facebook Open Graph, Flickr, Twitter, etc.). These APIs are generally based upon syntactic and structural representations (mainly JSON and XML) that are semantically deficient, making interoperability between social networks difficult. Other current efforts to unify Web 2.0 information are not semantic-based (ActivityStreams, Atom), further impeding efforts to automate information reasoning and fusion. Scattered attempts to semantically formalize social information via various specifications (e.g. Friend Of A Friend (FOAF), SIOC, and GoodRelations) have failed to unify the collective set of social/semantic community requirements.

New social sites and API versions pop up regularly. A unifying framework is desirable to unify and bridge social media models in a decentralized way, and inter-operate with other new and emerging social networks. To fill this gap, Image Matters created a family of micro-theories[4] for social networks called SocialML.

SocialML bridges this gap by providing a set of core ontologies for representing Social Networks, Persona, social activities and objects, organizations, social relationships and social network analysis metrics. It is clear that SocialML ontologies have the potential to have a tremendous impact on the interoperability of existing Web 2.0 social networks, as well as their integration with the Linked Data Web (Web 3.0). As a standard, SocialML

---

[4] http://2012books.lardbucket.org/books/sociology-brief-edition-v1.0/s04-03-theoretical-perspectives-in-so.html

would benefit the Web 2.0 community at large by removing the barriers of interoperability between various social media and social network services.

10.3 Portrayal Ontologies

Portrayal Ontologies specify a conceptual model for portrayal data, in particular symbols and portrayal rules. Portrayal rules associate features with symbols for the portrayal of the features on maps and other display media. These ontologies include classes, attributes and associations that provide a common conceptual framework that specifies the structure of and interrelationships between features, portrayal rules and symbols. It separates the content of the data from the portrayal of that data to allow the data to be portrayed in a manner independent of the dataset. The graphic description is intended to be format independent but convertible to any target formats (SVG, KML). The ontologies are derived from concepts found in existing portrayal specifications (ISO 19117, OGC Symbology Encoding and Styled Layer Descriptor Profile of WMS).

To favor reusability, the Portrayal ontologies are decomposed into four micro-theories (see Figure 6):

- **Style ontology**: defines the concept of Style and portrayal rules.
- **Symbol ontology**: defines the concept of SymbolSet and Symbol and structural definition of Symbol components.
- **Graphic Ontology**: defines graphic elements including graphic objects and attributes.
- **Portrayal Catalog Ontology**: Defines the concept of Portrayal Catalog

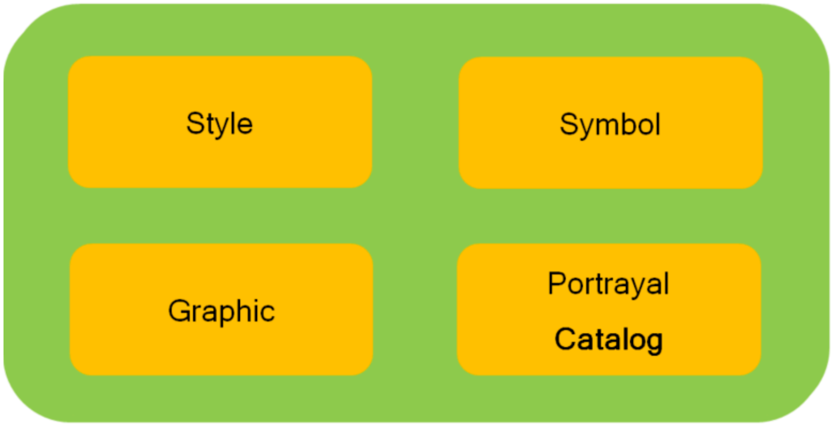


Figure 6 Portrayal Microtheories

More information about the portrayal ontologies can be found in the Symbology Mediation ER (OGC 15-058)

10.4 SPARQL Extensions Ontology

The SPARQL Extensions ontology defines concepts that allow the definition of custom functions and mapping using SPARQL standards. This ontology provides the core building blocks for defining semantic mediation mappings but it has also other applications such as defining constraints on classes, annotating classes with inference rules, defining templates for pre-canned queries. More information about the ontology can be found in the OGC Symbology Mediation Engineering Report (OGC 15-058)

10.5 Semantic Mediation Ontology

The Semantic Mediation Ontology defines the notion of **Alignment** between two ontologies. It leverages the SPARQL Extensions ontology by referring to Mapping instances. We may consider in the future migrating **Mapping** and **MappingType** concepts into this ontology as they are building block for semantic mapping. More information about the ontology can be found in the Symbology Mediation ER (OGC 15-058)

11 Review of NEO

11.1 Overview

During Testbed 11, NGA asked Image Matters to review the NSG Enterprise Ontology (NEO) developed by MITRE. The goal of the ontology is to describe semantically the **NSG Entity Catalog (NEC)** and **NSG Feature Data Dictionary (NFDD)** using Linked Data standards (RDF, OWL and SKOS). NEO was built according the rules defined in the ISO 19150-2 standard. NEO is composed of multiple OWL ontologies describing the National System for Geospatial Intelligence (NSG) Feature Data Dictionary (NFDD) and the NSG Application Schema (NAS).

The NSG Application Schema (NAS) - Part 1 - specifies an NSG-wide logical model for geospatial data that is technology neutral. This ***Platform Independent Model*** determines the syntactic structure used to represent the semantics specified by the **NSG Entity Catalog (NEC).** NEC conforms to ISO 19109, *Geographic information - Rules for application schema*, and its conceptual schema. It integrates conceptual schemas from multiple ISO 19100-series standards for geospatial information modeling, such as those for features, events, names and coverages (*e.g.*, grids, rasters, and TINs). It leverages and integrates geospatial information modeling practices from multiple community models (*e.g.*, MGCP, AIXM, MIDB, ENC, AML, and others) whose data are used and exchanged by NSG component systems. The NAS entities are organized into logical subject matter **Views (205) and View Groups (42)** for better searching and discovery capabilities by subject matter experts.

The **National System for Geospatial Intelligence (NSG) Feature Data Dictionary (NFDD)** specifies an NSG-wide data element dictionary for geospatial data. This dictionary includes feature concepts, attribute concepts with their domain types, and accompanying metadata. It conforms to a subset of ISO 19126, *Geographic information - Feature concept dictionaries and registers*, and its information schema. It draws upon multiple community dictionaries (e.g., Digital Geospatial Information Working Group Feature Data Dictionary (DFDD), Aeronautical Information Exchange Model (AIXM), Modernized Integrated Database (MIDB), IHO S-57, NATO Additional Military Layers (AML), and others) to specify an integrated feature data dictionary tailored to the requirements of the US DoD/IC.

The NEO ontology files are summarized in Table 2:

Table 2 NEO Ontology files

File name	Publication date	Description
irCodelist_2015-02-15.owl	02/15/2015	Ontology defining concept scheme and concept subclasses
nas_6.2.owl	03/06/2015	Ontology for NAS
nfdd_6.2.owl	03/06/2015	Ontology for NFDD
irCodelist_2015-02-15_skos.rdf	02/15/2015	Instance of IRCodeList
nas_6.2_skos.rdf	02/15/2015	SKOS Encoding of code lists in NAS
nfdd_6.2_skos.rdf	02/15/2015	SKOS Encoding of code lists in NFDD

11.2 Issues

11.2.1 Issues related to Controlled vocabularies

The NAS and NFDD are composed of a large number of code lists, which defines a list of permissible value (code) for a given value domain. ISO 19150-2 requires that code lists being encoded as subclass of **skos:ConceptScheme** and codes as subclass of **skos:Concept**. A number of errors were found in the encoding of the code lists that can be easily fixed. We illustrate some of these errors in the following listings.

The definition of the code class should be a subclass of **skos:Concept**, not skos:ConceptScheme.

```
<owl:Class rdf:about="&nas;/WaterMovementDataLocationReferenceWaterLevelType">
	<skos:prefLabel>Water Movement Data Location Reference Water Level Type</skos:prefLabel>
	<skos:definition>A coded domain value denoting the reference water level type of a water movement data location.</skos:definition>
	<iso19150-2:isEnumeration rdf:datatype="xsd:boolean">true</iso19150-2:isEnumeration>
	<rdfs:subClassOf rdf:resource="&skos;Concept~~Scheme~~"/>
	<rdfs:subClassOf rdf:resource="&nas;DataType"/>
	<dc:source>http://geo.aitcnet.org/NSGREG/as/view?i=103658</dc:source>
 </owl:Class>
```

The instantiation of the code type is wrongly generated as it adds the value on the class name, thus creating a new type for each value. The concept instances are missing reference to the concept schema instance. If multiple code notation exists for a code, the ontology should define a datatype for the **skos:notation** property value for each code system.

```
<nas:WaterMovementDataLocationReferenceWaterLevelType~~_highTide~~ rdf:about="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme/highTide">
	<skos:topConceptOf rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme"/>
	<skos:notation>1</skos:notation>  (create a datatype for the notation)
	<skos:prefLabel><![CDATA[Water Movement Data Location Reference Water Level Type : High Tide]]></skos:prefLabel>
	<skos:definition>The highest water level achieved during a tidal cycle.</skos:definition>
	 <skos:inScheme rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme"/>  (to add)
	<dc:source>http://geo.aitcnet.org/NSGREG/as/view?i=114122</dc:source>
   </nas:WaterMovementDataLocationReferenceWaterLevelType~~_highTide~~>
   <nas:WaterMovementDataLocationReferenceWaterLevelType~~_lowTide~~ rdf:about="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme/lowTide">
	<skos:topConceptOf rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme"/>
      <skos:inScheme rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme"/>  (to add)
	<skos:notation>2</skos:notation> (create a datatype for the notation)
```

```
    <skos:prefLabel><![CDATA[Water Movement Data Location Reference Water Level Type : Low Tide]]></skos:prefLabel>
    <skos:definition>The lowest water level achieved during a tidal cycle.</skos:definition>
    <dc:source>http://geo.aitcnet.org/NSGREG/as/view?i=114123</dc:source>
  </nas:WaterMovementDataLocationReferenceWaterLevelType~~_lowTide~~>
```

In order to support classification of ConceptScheme, we should introduce a subclass of ConceptScheme for each code list and then create an instance of this subclass to encode the code list instance.

```
  <skos:ConceptScheme rdf:about="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme">  (could create a subclass of ConceptScheme to support classification )
    <skos:prefLabel>Water Movement Data Location Reference Water Level Type - Concept Scheme</skos:prefLabel>
    <skos:definition>A coded domain value denoting the reference water level type of a water movement data location.</skos:definition>
    <dc:source>http://geo.aitcnet.org/NSGREG/as/view?i=103658</dc:source>
    <dcterms:isFormatOf rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType"/>
    <skos:hasTopConcept rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme/highTide"/>
    <skos:hasTopConcept rdf:resource="&nas;/WaterMovementDataLocationReferenceWaterLevelType_ConceptScheme/lowTide"/>
  </skos:ConceptScheme>
```

11.2.2 Issues related to Ontology

11.2.2.1 Feature classification

The Feature type defined in the NAS and NFDD ontologies does not subclass **Feature** class. Instead it uses the annotation property *iso19150-2:isFeatureType* with a Boolean value *true*. We suggest that each feature class in NAS and NFDD be a subclass of **geosparql:Feature**.

```
<owl:Class rdf:about="&nfdd;/AcousticSensor">
    <rdfs:subclasssOf rdf:resource="geosparql:Feature"/> (TO ADD)
    <skos:prefLabel>Acoustic Sensor</skos:prefLabel>
    <skos:definition>A sensor that is designed to detect pressure waves made by the propagation of a sound vibration(s) in gases, liquids, or solids. [desc] Acoustic sensors are either passive (only receiving pressure wave pulses) or active (producing their own
```

```
output pressure wave signal and detecting the reflection of that wave's energy from a
physical body in the water). Acoustic sensors are used in submarines, on surface ships
for anti-submarine warfare, and on research vessels conducting hydrographic survey
operations. Acoustic sensors are also used in the determination of the depth of water by
maritime vessels while navigating.</skos:definition>
        <iso19150-2:isFeatureType rdf:datatype="xsd:boolean">true</iso19150-
2:isFeatureType>
        <dc:source>http://geo.aitcnet.org/NSGREG/fdd/view?i=1817643</dc:source>
 </owl:Class>
```

11.2.2.2 UML versus OWL Modeling

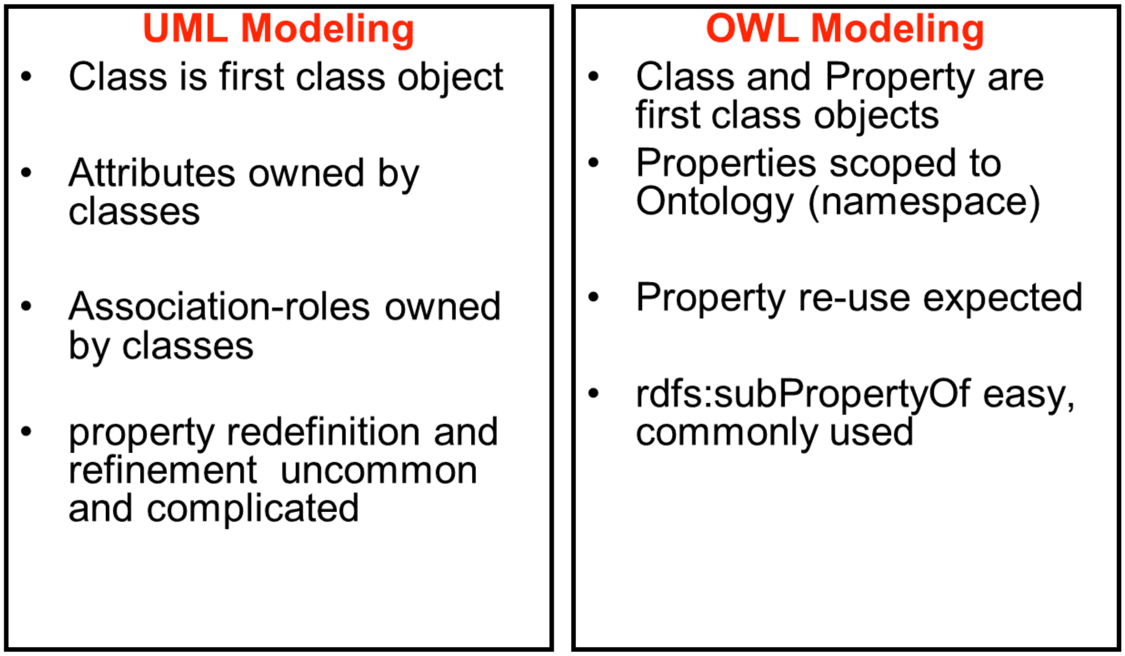
The NEO ontologies were derived from object-oriented UML models. Modeling ontologies is very different than modeling object oriented software artifacts. This section attempts to outline the differences and similarities between UML modeling and OWL modeling.

UML is a notation originally developed for modeling object oriented software artifacts. The OGC and other standards organizations also use UML for documenting content models. OWL is a notation for knowledge representation. These are two quite different problems so asking about expressivity differences is going to lead you into apples versus oranges comparison. In particular thinking in object oriented terms when working with OWL or RDFS will almost always lead you astray.

To take an example, both OWL and UML have things called "classes" and those classes are related in some sort of hierarchy that you can depict via a diagram. However, they are not the same notion at all.

In UML a class is a *software object* acting as a *template (or frame)* for objects. You can create instances of a class and that creation process has a procedural semantics involving things like assigning values to attributes (aka slots, members). So procedural notions, such as a default value for an attribute which is resolved at construction time, is a simple well-defined thing in UML. Instance objects also have some associated storage so UML distinguishes between containment and association - things are that are stored in the object v. things that are outside the object. Instance objects have a runtime semantic so you can have notions of static values and mutable values. Attributes and association role are owned by classes, i.e. their existence depends on the existence of the classes. They are not first class modeling concepts. Asserting a property on a class (or discovering such an assertion) cannot lead to inferring that it is a member of further classes (for example the fact that a class has property length does not lead the class is a spatial object).

None of that is true in RDFS or OWL or similar languages. In OWL modeling a class is a *category*, it is a label given to a set of things in a domain. In OWL terms a class is simply the set of things which are members of that set. So RDFS/OWL has no notion of instantiating or constructing an instance and so no notion of default values for instance creation. If you have a resource and it meets the criteria for membership of the class then it is a member of that class; if you didn't already know that your resource was in that class then you can derive this information by reasoning. Asserting a property on a resource (or discovering such an assertion) can lead to you inferring that it is a member of further classes. Resources (in RDFS terms, Individuals in OWL terms) are not things with state, storage or runtime semantics. They are simply identifiers for things in your domain. Assertions about those resources can be made, found or derived but the resources themselves aren't objects with slots and so notions like static and public/private have no meaning here. This is not a limitation of OWL or RDFS so much as the fundamental nature of what you are modelling. OWL classes are like labels for concepts, UML classes are like templates that define a runtime object and its storage.



UML Modeling

- Class is first class object
- Attributes owned by classes
- Association-roles owned by classes
- property redefinition and refinement uncommon and complicated

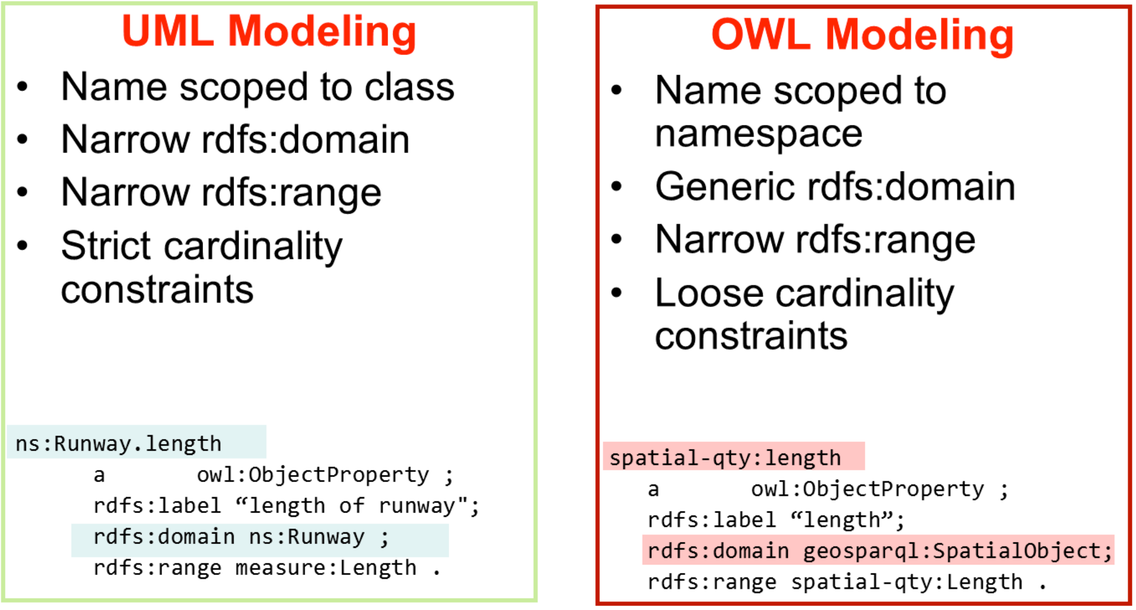
OWL Modeling

- Class and Property are first class objects
- Properties scoped to Ontology (namespace)
- Property re-use expected
- rdfs:subPropertyOf easy, commonly used

Figure 7 UML Modeling versus OWL modeling

In addition in OWL, properties are first class objects and are scoped to the ontology (namespace). Property re-use in different classes is expected and commonly used as well as specialization of properties (using subPropertyOf hierarchy). Property redefinition and refinement is very uncommon and complicated to model in UML. Figure 7 and Figure 8 summarizes the difference between UML modeling and OWL Modeling.

In UML the domain and range of the properties are very narrow as the properties belong by default to the Class, thus the property has a default scope to the class. Additional annotations are needed to indicate that the properties are global (as defined in ISO 19150-2). In OWL properties are typically defined globally and thus have a broader domain and narrow range. In addition cardinality of OWL properties are looser that cardinality in UML model. Figure 8 shows an example where the length of a runway property has a domain Runway. In OWL the length can defined globally and has broader domain SpatialObject. The conversion from UML to OWL using the default settings generates ontology **very limited expressiveness and reusability**. In OWL, reuse of external vocabularies is commonly expected. The choice of the vocabularies requires the expertise of an ontologist and subject-matter expert and cannot be automated.



UML Modeling

- Name scoped to class
- Narrow rdfs:domain
- Narrow rdfs:range
- Strict cardinality constraints

```
ns:Runway.length
       a       owl:ObjectProperty ;
       rdfs:label “length of runway";
       rdfs:domain ns:Runway ;
       rdfs:range measure:Length .
```

OWL Modeling

- Name scoped to namespace
- Generic rdfs:domain
- Narrow rdfs:range
- Loose cardinality constraints

```
spatial-qty:length
    a       owl:ObjectProperty ;
    rdfs:label “length”;
    rdfs:domain geosparql:SpatialObject;
    rdfs:range spatial-qty:Length .
```

Figure 8 Property modeling in UML versus OWL

Despite all these differences there is however some connection between both modeling languages. Both UML and OWL are languages for modelling. They are modelling totally different things and so have different capabilities and a completely different approach to semantics but there are some structural similarities. MOF (Meta Object Facility) is the meta-modelling tool that underlay UML. It is a language in which you can express other modelling languages. So UML is specified in MOF. You can do the same for RDFS and OWL - that is, express their metamodels in MOF. This is what ODM (Ontology Definition Metamodel) provides which provides a profile for writing RDF and OWL within UML. It also includes mappings between UML and OWL as well as mappings amongst RDF, RDFS, Common Logic and Topic Maps.

This discussion should highlight why modeling ontologies cannot be done by simply converting (automatically or not) UML diagram to OWL. Ontologies are better designed from the ground-up but can be informed by concepts expressed in UML models.

11.2.2.3 Lack of reuse of existing vocabularies

In order to make it as easy as possible for client applications to process NEO data, the ontology should reuse terms from well-known vocabularies wherever possible. New terms should be defined only when required terms cannot be found in existing vocabularies. Using an automated process to generate ontology from UML model goes against this best practice.

11.3 Recommendations

The approach taken in NEO to convert code lists as taxonomies is sound and can be fully automated following the guidelines of ISO 19150-2. However the ontology for NFDD and NES requires a careful choice of micro-theories such as the geospatial micro-theories contributed to OGC by Image Matters and the ones currently under work in the Spatial Data on the Web OGC-W3C Working Group. The reuse of well-established vocabularies would facilitate the processing of NEO by software agents that understand these core ontologies. This work requires subject-matters expertise to carefully choose the micro-theories and carefully define the sub-properties hierarchies (for example for physical properties). Annotating existing UML models with these additional semantic information with the aim to automate the conversion to OWL could be achieved, but the cost of doing it will not outweigh the benefit of doing it using a manual process.

12 OGC Web Services

12.1 Service Oriented Architecture (SOA)

A Service Oriented Architecture (SOA) is based on the concept of a **service**. Depending on the service design approach taken, each SOA service is designed to perform one or more activities by implementing one or more service operations. As a result, each service is built as a discrete piece of code. This makes it possible to reuse the code in different ways throughout the application by changing only the way an individual service interoperates with other services that make up the application, versus making code changes to the service itself. SOA design principles are used during software development and integration.

SOA generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. For example, several disparate departments within a company may develop and deploy SOA services in different implementation languages; their respective clients will benefit from a well-defined interface to access them.

SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. Rather than defining an API, SOA defines the interface in terms of protocols and functionality. An endpoint is the entry point for such a SOA implementation.

12.2 Resource Oriented Architecture

Resource Oriented Architecture is a specific set of guidelines of an implementation of the REST-style architecture. According Leonard Richardson and Sam Ruby in their book entitled 'RESTful Web Services',

ROAs are based on four concepts:

- Resources (e.g. the article about REST in the Wikipedia).Their names (URIs).
- The URI is the name and address of a resource. For example,http://www.wikipedia.org/wiki/Representational_State_Transfer.
- Their representations. A resource is a source of representations.
- The links between them. Normally a hypermedia representation of a resource contains links to others resources.

and four properties:

- **Addressability**. Addressable applications expose a URI for every piece of information they might conceivably serve.
- **Statelessness**. Statelessness means that every HTTP request happens in complete isolation. The server never relies on information from previous requests.
- **Connectedness**. A Web service is connected to the extent that you can put the service in different states just by following links and filling out forms.
- **A uniform interface**. In ROAs, HTTP is the uniform interface. GET method to retrieve a representation of a resource, PUT method to a new URI or POST method to an existing URI to create a new resource, PUT method to an existing URI to modify a resource and DELETE method to remove an existing resource. Probably HTTP methods are not a perfect interface but what is important is the uniformity. The point is not that GET is the best name for a read operation, but that GET means “read” across the Web. Given a URI of a resource, everybody knows that to retrieve the resource s/he has to send a GET request to that URI.

Since the SOA definition is independent of the technical architecture of the services it encompasses all REST/HTTP applications. ROA can be seen as a term to describe that part of a SOA implemented following the guidelines stated above. That is, ROA is less general than SOA since it is not independent of the technical architecture of the services. The term ROA is often used to emphasize that such an architecture in based on HTTP objects that respond to one or more of the standard HTTP methods. Why? Because SOA has been traditionally focused on interfaces and when people talk about interfaces they tend to use terms like “method”, “operation”, etc. which are strongly related to the RPC-style. Thus, to avoid misunderstandings the term ROA is used to make clear that we are talking about REST-style architectures.

12.3 Possible approaches to semantic enablement

During the testbed, we identified three different approaches for the semantic enablement of the OGC Services.

12.3.1 Extension of existing services protocol

Existing OGC services can be extended to support Linked Data representation as long as the services are RESTful. For example, SensorThing provides a RESTful API to access observations and sensor information. It is relatively easy to extend the API by adding an additional RDF representation for each resource endpoint without breaking the existing APIs. If the JSON model of the service is compatible with the ontology representing the data model, it is possible to use JSON-LD by adding a JSON-LD context to existing JSON response. This allows clients to consume JSON-LD document and convert the information to RDF representation for further processing.

12.3.2 Wrapper approach

A large numbers of Web applications have started to make their data available on the Web through Web APIs. Examples of data sources providing such APIs include Social Media APIs (Twitter, YouTube, Flickr), OGC Services such as WFS, WMS, WCS, SOS,. Different APIs provide diverse query and retrieval interfaces and return results using a number of different formats such as XML, JSON or ATOM. This leads to three general limitations of Web APIs:

- their content cannot be crawled by search engines
- Web APIs cannot be accessed using generic data browsers
- Mashups are implemented against a fixed number of data sources and cannot take advantage of new data sources that appear on the Web.

These limitations can be overcome by implementing Linked Data wrappers around APIs. In general, Linked Data wrappers do the following:

- They assign HTTP URIs to the non-information resources about which the API provides data.
- When one of these URIs is dereferenced asking for application/rdf+xml, the wrapper rewrites the client's request into a request against the underlying API.
- The results of the API request are transformed to RDF and sent back to the client.

This approach has been used to semantic enabled Social Media APIs by using a semantic wrapper around APIs. We also use this approach for semantically enabled WFS-G by transforming GeoSPARQL queries to OGC filter queries on the fly and converting the GML response to Linked Data representation.

12.3.3 New RESTful Service API

The wrapper approach sometimes may lead to performance problems due to the overhead of conversion from one representation to another or limitations of the wrapped API. When it is the case, it is preferable to design RESTful API that eases the implementation and access of information with the lowest overhead.

Most of the existing OGC Services are not RESTful. Because Linked Data relies on REST principles, it is easier to define a RESTful API for existing OGC services and provides Linked Data representation associated for each resource of the services. The Linked Data Best practices published by W3C and ROA guidelines should be used to design the new RESTful API. The use of the Linked Data Platform API should be used when applicable as well as the use of JSON-LD to bridge the gap between Linked Data and web developers using JSON. The use of resolvable URIs has shown direct benefit into integrating and linking multiples sources.

13 Semantic Services implemented for Testbed 11

13.1 Semantic Mediation Service

The Semantic Mediation Service is a new service introduced during this testbed to support the Symbology Mediation CCI subthread. The mediation services addresses the first task of performing the mediation of the information represented by the symbology. The Semantic Mediation Service (SMS) can be reused for different contexts when alignment between one ontology and another is needed. For example, it can be used for search information expressed in one ontology to find information expressed in a different ones. Future extensions may support SPARQL query rewriting for a given alignment.

The SMS was designed to be RESTful and to use Linked Data standards. Figure 9 describes the architecture of the service. The service is composed of two graph stores: the first one contains the definition of the alignments, mapping definitions. The second store contains the definition of the functions and mapping types. The mapping engine is used to perform the transformation from one Linked Data Model to another one. It leverages the GeoSPARQL engine that is augmented with the plugins functions and rules. The Service exposed a REST API to access the concepts from the knowledge stores using a Linked Data API. It provides also a Semantic Mediation REST API that performs the mediation work for a specific alignment. At last it provides a GeoSPARQL endpoint capable to query the Alignment database.

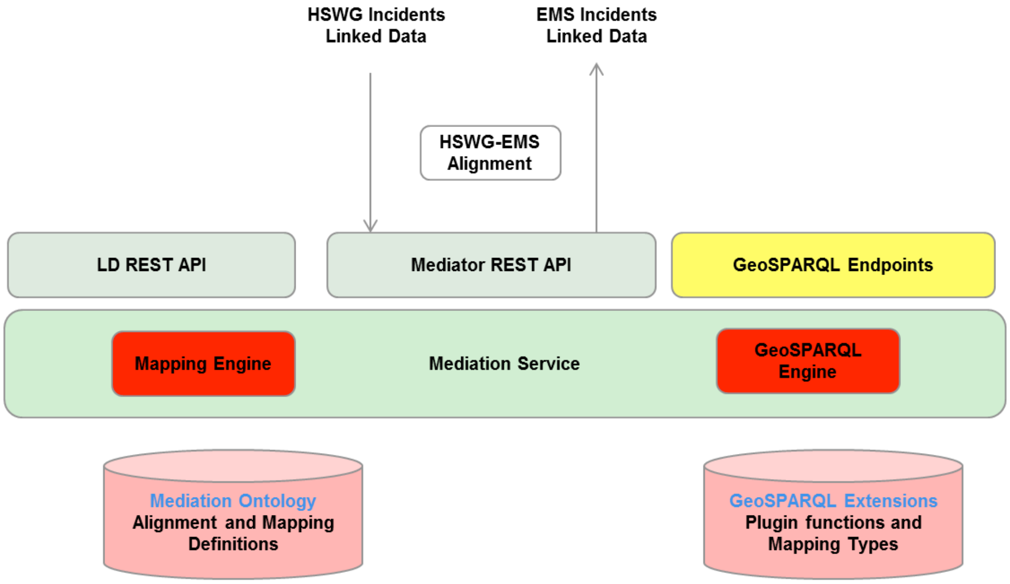


Figure 9 Semantic Mediation Service Architecture

13.2 Semantic Portrayal Service

During the testbed, Image Matters deployed an initial version of semantic portrayal service (also known as symbology service) online at the following endpoint: http://ows.usersmarts.com/portrayal/api

The server has been loaded with the EMS, HSWG symbols, taxonomies and portrayal rules produced during the testbed. The service has also an initial REST API to fetch symbols and symbol sets in TTL, RDF/XML, JSON-LD. N3 and NT formats.

The server consists of a standard RDF database (Systap BlazeGraph) where all the portrayal information was stored as Linked Data. A REST API was built on top of the repository.

The main endpoint is the SPARQL endpoint allowing to access any portrayal information (style, portrayal rules, symbol sets, symbols, graphics and supporting taxonomies). Two other endpoints provides a Linked Data API to access symbol sets and symbols. Future extensions will provide access to styles, portrayal rules and graphics in a RESTful way.

13.3 Semantic Gazetteer Services

For this testbed, Image Matters deployed two semantic gazetteers to support gazetteer linking with National Hydrology Dataset (NHD). Gazetteer Linking is based on the premise that features in a gazetteer and other sources of information have already been matched and the match between identifiers is stored in a concordance or is embedded in a data source. In the case of NHD, the GNIS identifier was stored with the flowlines and gauges information.

To take full advantage of the semantic web and ability to quickly move across links, we encoded the NHD and GNIS gazetteers data set using RDF. The goal of the task was to encode information in RDF from gazetteers and NHD by leveraging existing infrastructure (WFS-Gs, RDBMS) using semantic mapping components (mapping spatial RDBMS to RDF), demonstrate a capability to link new information available from related resources (obtaining information from sources at least two sources distant from the original source), query and select the information of interest, and return the information in a query. This was done using open linked data standards (RDF,OWL,SPARQL) and the OGC GeoSPARQL query.

13.3.1 Semantic Mapping components

Until recently, data integration has been accomplished using a single layer approach by writing a data product translator from one format to another. For example, it is common practice today to use XSLT to transform one XML document to another XML format. The problem with this approach is that it mixes the structural and semantic transformation together. Also it does not scale, because it is based on a N-to-N mapping approach, and is error-prone due to reliance on human interpretation of data products.

The rules, which carry out the complete transformation process in one shot, have proven to be very complex. This causes serious problems in implementing and maintaining the rules of transformation. These problems arise due to the mixture of several different aspects of the overall transformation process, such terminology, granularity representation and structural and syntactic alignment. For this reason, any re-use of such rules is practically impossible.

To overcome this bottleneck a multi-layered framework should be used, which separates different aspects of the transformation process. The approach used in Image Matters Knowledge Mapping Service (KMS) is able to transform a complex programming task into a simple plug-and-play process where straightforward rule patterns are selected, instantiated, and combined. KMS uses a methodology for data integration based on a three-layer model, as presented in the Figure 10. The model contains a Data Product layer, a Data Model layer, and an Ontology layer.

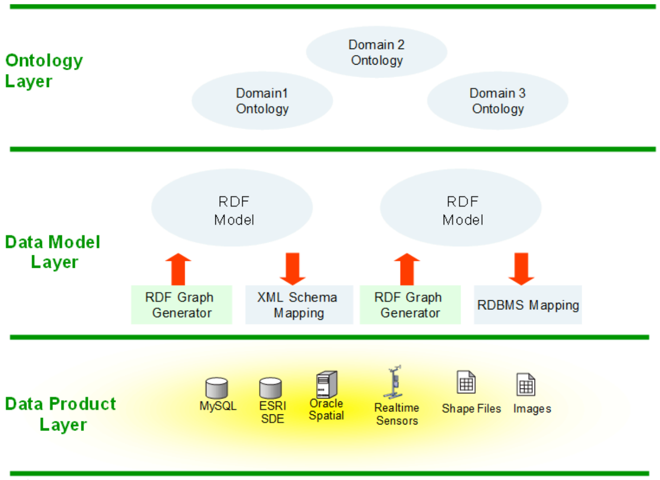


Figure 10 Three layer approach for data integration

KMS provides the ability to map ‘legacy’ (geospatial or not) data stores and formats to a RDF knowledge representation using a unified declarative mapping expressed in RDF. KMS uses this mapping to translate semantic query (graph query, SPARQL,...) to native query language (such as Spatial SQL, XPath/XQuery, OGC Filter) or API calls. This framework allows the virtualization of the data into a semantic graph representation and provides real-time access to data into a unified semantic representation, which could be leveraged by other knowledge-centric service components (reasoners, query engine, (Geo)SPARQL endpoints, semantic mediation, visualizations).

For this testbed, Image Matters investigated the semantic mapping of the database dump from Geonames.org, USGS GNIS gazetteers and NGA Geonames. The semantic Mapping component was used to offer virtual GeoSPARQL endpoints over the mapped database and services. This approach provided a unified knowledge representation, query language and protocol to access existing gazetteer data infrastructure as illustrated in Figure 11. In the OWS-10, semantic mapping was used to integrate WFS-G serving NGA Geonames (Interactive Instruments) and USGS GNIS (Compusult). However the semantic mapping was limited due to some limitations in OGC Filter and issues related to usage of XLink for complex features, which tremendously impacted performance (issues are explained more in details in the OGC Engineering Report OGC 14-029.

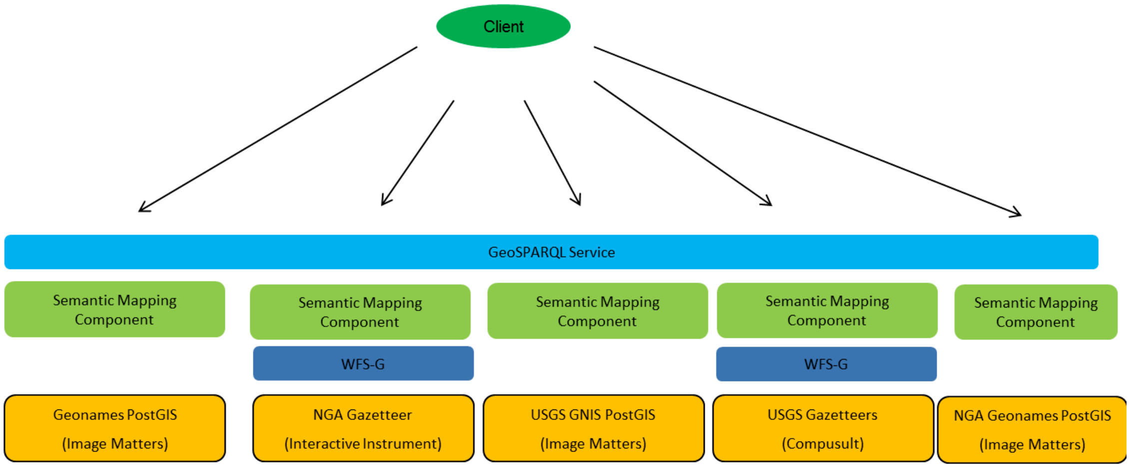


Figure 11 Semantic Gazetters Integration

13.3.2 Geonames

For this project, a database dump of Geonames was installed and indexed (spatial and text) in a PostGIS database instance on an Image Matters Server. KMS Semantic mappings from relational database to RDF dataset were defined. Such mappings provide the ability to view existing relational data in the RDF data model, expressed in a structure and target vocabulary (ontology) aligned with the ISO 19112 model. The mappings are themselves RDF graphs and written down in Turtle syntax. The KMS processor was adapted to support directly geospatial functions defined in GeoSPARQL specification.

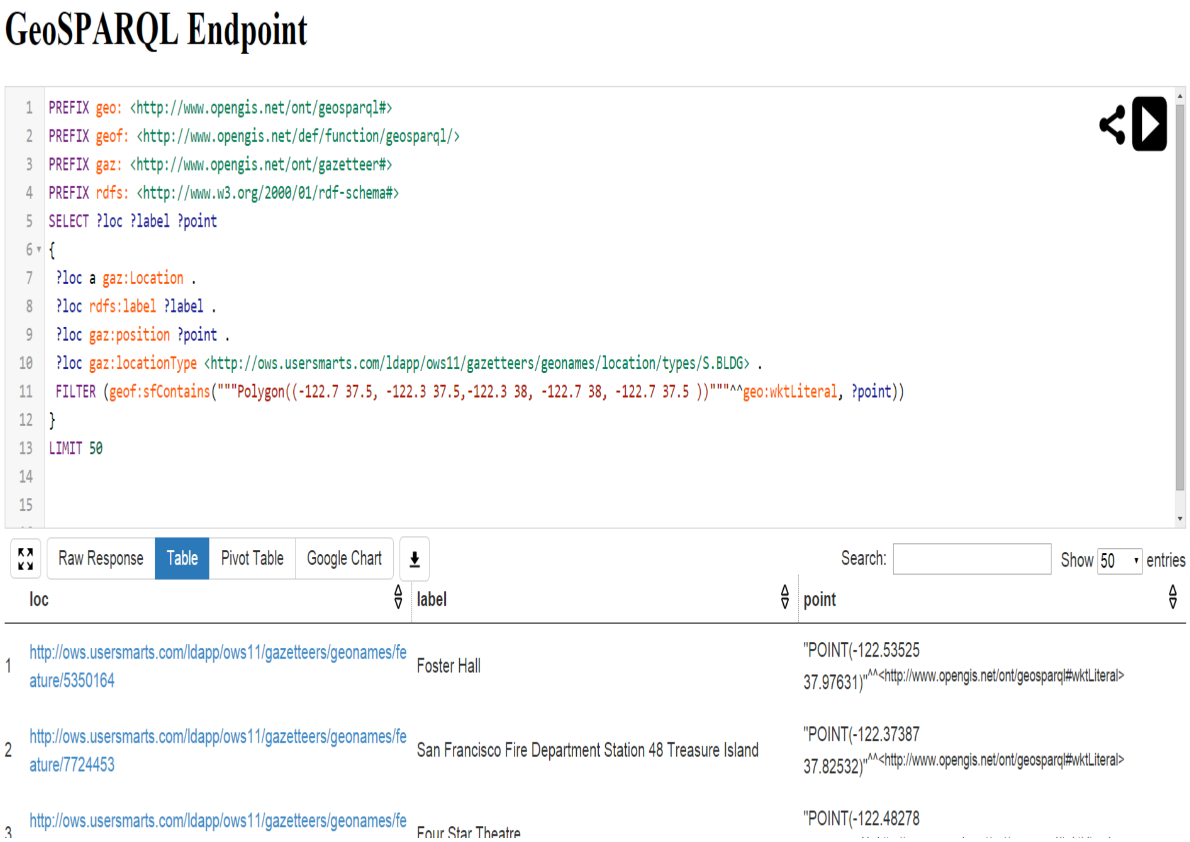
13.3.2.1 GeoSPARQL endpoint

The database was made accessible through a GeoSPARQL endpoint at the following address:

http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/sparql

GeoSPARQL queries sent to the server are translated to one or more spatial SQL queries and results are converted on the fly to Linked Data. Using this approach performance of the system was similar to the native query as the overhead consists mainly to query rewriting and serialization of the final result into Linked Data representation.

For the OWS 11 demonstration, most of the URLs of the resources in Geonames were **made resolvable and accessible through a REST API**. The data can be returned in RDF, TTL, N3, NT, or JSON-LD format. The SPARQL endpoint provides a HTML-based client with syntax validation and allows the visualization of the results in raw format, table or visual view (charts, maps) when the structure of the table is compatible with the view (see Figure 12).



GeoSPARQL Endpoint

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX gaz: <http://www.opengis.net/ont/gazetteer#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?loc ?label ?point
{
 ?loc a gaz:Location .
 ?loc rdfs:label ?label .
 ?loc gaz:position ?point .
 ?loc gaz:locationType <http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/location/types/S.BLDG> .
 FILTER (geof:sfContains("""Polygon((-122.7 37.5, -122.3 37.5,-122.3 38, -122.7 38, -122.7 37.5 ))"""^^geo:wktLiteral, ?point))
}
LIMIT 50
```

	loc	label	point
1	http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/feature/5350164	Foster Hall	"POINT(-122.53525 37.97631)"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
2	http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/feature/7724453	San Francisco Fire Department Station 48 Treasure Island	"POINT(-122.37387 37.82532)"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
3	http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/fe	Four Star Theatre	"POINT(-122.48278

Figure 12 GeoSPARQL Endpoint for Geonames

Follows the descriptions of some of the endpoints of the Linked Data REST API for Geonames gazetteer.

13.3.2.2 Feature endpoint:

Example: Get the Linked Data description feature id 9688452 (San Francisco Police Department Richmond District)

http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/feature/9688452

This endpoint returns a Turtle description of the feature:

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/feature/9688
452>
        a       <http://www.opengis.net/ont/gazetteer#Location> ,
<http://www.opengis.net/ont/spatial#SpatialThing> ,
<http://www.opengis.net/ont/geosparql#SpatialObject> ,
<http://www.opengis.net/ont/geosparql#Feature> ;
        <http://www.w3.org/2000/01/rdf-schema#label>
                "San Francisco Police Department Richmond District" ;
        <http://www.geonames.org/owl#fclass>
```

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/location/typ
es/S> ;
        <http://www.geonames.org/owl#geonameId>
                "9688452" ;
        <http://www.geonames.org/owl#population>
                0 ;
        <http://www.geonames.org/owl#timeZone>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/timeZone/Ame
rica/Los_Angeles> ;
        <http://www.opengis.net/ont/gazetteer#admin1>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/admin1/US.CA
> ;
        <http://www.opengis.net/ont/gazetteer#admin2>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/admin2/US.CA
.075> ;
        <http://www.opengis.net/ont/gazetteer#country>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/countries/US
> ;
        <http://www.opengis.net/ont/gazetteer#locationType>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/location/typ
es/S.BLDG> ;
        <http://www.opengis.net/ont/gazetteer#position>
                "POINT(-122.46448
37.77998)"^^<http://www.opengis.net/ont/geosparql#wktLiteral> ;
        <http://www.opengis.net/ont/spatial#partOf>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/admin1/US.CA
> ,
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/countries/US
> ,
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/admin2/US.CA
.075> .
```

The Linked Data can be returned in different formats using file extensions or content negotiation.

http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/feature/9688452.jsonld

http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/feature/9688452.rdf

Most of the links referred in the response were made resolvable (FeatureClass, Admin1)

13.3.2.3 Feature class Endpoint

FeatureClass S.BLDG has a LD description at the following endpoint.

http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/location/types/S.BLDG

The endpoint returns the following response:

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/location/typ
es/S.BLDG>
        a       <http://www.w3.org/2004/02/skos/core#Concept> ,
<http://www.opengis.net/ont/gazetteer#LocationType> ;
        <http://www.w3.org/2004/02/skos/core#:prefLabel>
                "building(s)" ;
        <http://www.w3.org/2004/02/skos/core#code>
                "S.BLDG" ;
        <http://www.w3.org/2004/02/skos/core#definition>
                "a structure built for permanent use, as a house,
factory, etc." ;
        <http://www.w3.org/2004/02/skos/core#inScheme>
                <http://www.geonames.org/owl#LocationTypeScheme> .
```

13.3.2.4 Admin1 endpoint

US.CA Administration level 1 has a linked data description at the following endpoint

http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/admin1/US.CA

The endpoint returns the following response

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/admin1/US.CA
>
        a       <http://www.opengis.net/ont/spatial#SpatialThing> ,
<http://www.opengis.net/ont/gazetteer#Location> ,
<http://www.opengis.net/ont/geosparql#SpatialObject> ,
<http://www.opengis.net/ont/geosparql#Feature> ;
        <http://www.geonames.org/owl#adminName>
                "California" ;
        <http://www.geonames.org/owl#code>
                "US.CA" ;
        <http://www.geonames.org/owl#externalLink>
                <http://en.wikipedia.org/wiki/California> ;
        <http://www.geonames.org/owl#geonameId>
                "5332921" ;
        <http://www.opengis.net/ont/gazetteer#locationType>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/location/typ
es/A.ADM1> ;
        <http://www.opengis.net/ont/spatial#partOf>
```

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/geonames/countries/US
> ;
        <http://www.w3.org/2004/02/skos/core#altLabel>
                "□□□□□□□□□□□□□□□□□□"@new , "Kalifornija"@bs ,
"Californie"@sco , "کالېفورنیا"@ps , "□□□□□□□□"@ja , "California"@es ,
"Kalifornien"@de , "Kalifornia"@eu , "Calif"@abbr , "Całifornia"@vec ,
"Kalifornien"@lb , "Калифорния"@bg , "Kalifornia"@pl ,
"Kalifornija"@hbs , "Kaliforniska"@hsb , "Kalifornien"@als ,
"Kaliforniya"@gag , "Kalifornija"@lt , "□□□□□□□□□□□□□□□□"@te ,
"□□□□□□□□□□□□□□□□□□□□"@bpy , "Калифорнија"@sr , "Kalifornia"@bi ,
"Kalėfuornėjė"@sgs , "kalifornias"@jbo , "□□□□□□□□□□□□□□□□□"@kn , "□□□□□
□"@ko , "California suyu"@qu , "Kalifornija"@hr , "Kaliforniya"@diq ,
"Calafòrnia"@gd , "کالیفرنیا"@mzn , "Kalifornia"@io , "كاليفورنيا"@arz
, "Kalifornia"@hu , "Kalifornia"@fo , "□□□□□□□□□□□□□"@ml ,
" , ckb@"کالیفۆرنیا"□□□□□□□□□□□□□□□□□"@sa , "□□□□□□"@iu ,
"Californien"@da , "California Republic"@en , "Kalifornio"@eo ,
"Califòrnia"@ca , "□□□□□□"@zh , "کیلیفـورنیـا"@pnb , "Калифорния"@sah ,
"Kalifornia"@se , "Калифорния"@kk , "Калифорния"@krc , "Kalifonän"@vo ,
"Kalifornia"@sk , "□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"@my , "Kâ-li-fuk-nì-
â"@hak , "Kalifornía"@is , "Califòrnia"@oc , "Califòrnia"@pms ,
"Kaliforniya"@uz , "Kalifornija"@sl , "□□□□□□□□□□□□□□□□□"@hi ,
"Калифорния"@ba , "كاليفورنيا"@ar , "Californië"@zea , "Калифорния"@ru
, "Калифорни"@mn , "Калифорни"@xal , "Kalifornija"@lv ,
"□□□□□□□□□□□□□□□□□□"@ne , "Калифорни"@mrj , "Штат Каліфорнія"@be ,
"Californie"@fr , "Καλιφόρνια"@el , "Kalifornia"@su , "Alta
California"@es , "CA"@abbr , "Калифорни"@cv , "Каліфорнія"@uk ,
"Karapōnia"@mi , "□□□□□"@wuu , "Kaliforniya"@tr , "Californeye"@wa ,
"Califórnia"@pt , "California suyu"@ay , "קליפורניה"@he ,
"California"@no , "Kalifornien"@sv , "□□□□□□□□□□□□□□□□□□□□"@bn ,
"Kalifornia"@sq , "Kalifornië"@af , "Kaliforniye Shitati"@ug ,
"Kaliforniya"@az , "□□□□□□□□□□□□□□□□□□□□"@th , "□□□□□□"@am , "کیلی
" , ur@"فورنیا□□□□□□"@gan , "Kalifornien"@bar ,
"□□□□□□□□□□□□□□□□□□□□□"@bo , "Kaliforni"@kw , "□□□□□□□□□□□□□□"@pa ,
"Kalifornia"@br , "Californië"@li , "Kalifornien"@nds , "Kalifornje"@fy
, "Kalifòni"@ht , "Golden State"@en , "□□□□□□□□□□□□□□"@ta ,
" , yi@"קאליפארניע"□□□□□□"@yue , "Калифорни"@os , "Estado de
California"@es , "Kalifọ́rníà"@yo , "Калифорния"@tt , "□□□□□□□□□□"@ka ,
"Kalifornia"@so , "Kalifornie"@cs , "Калифорний"@mhr ,
"Kaliforńijo"@szl , "Kalifornia"@mg , "Akééháshį́į́h Hahoodzo"@nv ,
"Califfornia"@cy , "Калифорнија"@mk , "□□□□□□□□□□"@hy , "Kaleponi"@haw
 , fa , "Kalifornia"@fi , "State of California"@en@"کالیفرنیا" ,
"Californië"@nl , "□□□□□□□□□□□□□□□□"@mr , "California"@en ,
"Kalifornien"@stq .
```

13.3.3 GNIS Gazetteers

We deployed two versions of the GNIS Gazetteers.

13.3.3.1 Version 1: PostGIS semantic mapping of GNIS

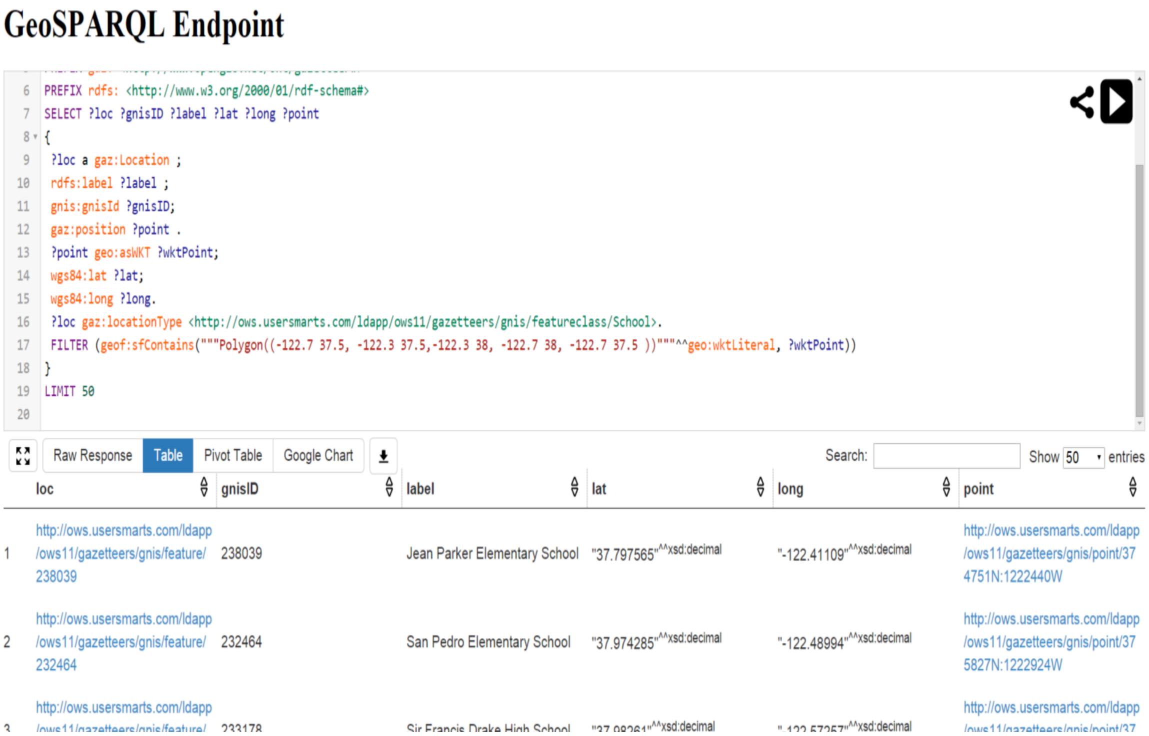
The first version is based on the mapping of PostGIS database containing the NationalFile for GNIS. It uses the same mapping techniques than Geonames, i.e. uses a declarative mapping from DB schema to OWL. The GeoSPARQL queries are converted to SQL queries and results are returned in RDF. For the OWS 11 demonstration, we made most of the URL of the resources in GNIS resolvable and accessible through REST API. The data can be returned in RDF, TTL, N3, NT, or JSON-LD format. The SPARQL endpoint provides a HTML-based client with syntax validation and allows to visualize the results in raw format, table or visual view when the data structure format. The GNIS mapping is richer than the second version based on WFS as most of the column of the GNIS are mapped to RDF.

The following are the endpoints for Semantic Geonames Gazetteer

SPARQL Endpoint

http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/sparql

Example of query using GeoSPARQL (Find Schools around San Francisco) is showed in Figure 13



GeoSPARQL Endpoint

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?loc ?gnisID ?label ?lat ?long ?point
{
 ?loc a gaz:Location ;
 rdfs:label ?label ;
 gnis:gnisId ?gnisID;
 gaz:position ?point .
 ?point geo:asWKT ?wktPoint;
 wgs84:lat ?lat;
 wgs84:long ?long.
 ?loc gaz:locationType <http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/featureclass/School>.
 FILTER (geof:sfContains("""Polygon((-122.7 37.5, -122.3 37.5,-122.3 38, -122.7 38, -122.7 37.5 ))"""^^geo:wktLiteral, ?wktPoint))
}
LIMIT 50

```

	loc	gnisID	label	lat	long	point
1	http://ows.usersmarts.com/ldapp /ows11/gazetteers/gnis/feature/ 238039	238039	Jean Parker Elementary School	"37.797565"^^xsd:decimal	"-122.41109"^^xsd:decimal	http://ows.usersmarts.com/ldapp /ows11/gazetteers/gnis/point/37 4751N:1222440W
2	http://ows.usersmarts.com/ldapp /ows11/gazetteers/gnis/feature/ 232464	232464	San Pedro Elementary School	"37.974285"^^xsd:decimal	"-122.48994"^^xsd:decimal	http://ows.usersmarts.com/ldapp /ows11/gazetteers/gnis/point/37 5827N:1222924W
3	http://ows.usersmarts.com/ldapp /ows11/gazetteers/gnis/feature/	233178	Sir Francis Drake High School	[illegible]^^xsd:decimal	[illegible]^^xsd:decimal	http://ows.usersmarts.com/ldapp /ows11/gazetteers/gnis/point/37

Figure 13 GeoSPARQL Endpoint for GNIS

Follows the descriptions of some of the endpoints of the Linked Data REST API for Geonames gazetteer.

Feature endpoint:

Example: Get the Linked Data description feature id 496185 (Leesburg,KY)

http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/feature/496185

The endpoint returns the following response in Turtle format

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/feature/496185>
        a       <http://www.opengis.net/ont/geosparql#SpatialObject> ,
<http://www.opengis.net/ont/spatial#SpatialThing> ,
<http://www.opengis.net/ont/geosparql#Feature> ,
<http://www.opengis.net/ont/gazetteer#Location> ;
        <http://www.w3.org/2000/01/rdf-schema#label>
                "Leesburg" ;
        <http://purl.org/dc/terms/created>
                "1979-09-20" ;
        <http://www.opengis.net/ont/gazetteer#admin1>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/admin1/KY> ;
        <http://www.opengis.net/ont/gazetteer#admin2>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/admin2/KY.097> ;
        <http://www.opengis.net/ont/gazetteer#country>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/countries/US> ;
        <http://www.opengis.net/ont/gazetteer#locationType>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/featureclass/Pop
ulated Place> ;
        <http://www.opengis.net/ont/gazetteer#position>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/point/381740N:08
42506W> ;
        <http://www.opengis.net/ont/geosparql#asWKT>
                "POINT(-84.4183
38.2945)"^^<http://www.opengis.net/ont/geosparql#wktLiteral> ;
        <http://www.opengis.net/ont/spatial#partOf>

<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/countries/US> ,
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/admin1/KY> ,
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/admin2/KY.097> ;
        <http://www.opengis.net/testbed11/ont/gnis#gnisId>
                "496185"^^<http://www.w3.org/2001/XMLSchema#int> ;
        <http://www.w3.org/2003/01/geo/wgs84_pos#lat>
                38.294518 ;
        <http://www.w3.org/2003/01/geo/wgs84_pos#long>
                -84.418274 .
```

The Linked Data can return different formats using file extensions or content negotiation.

http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/feature/496185.jsonld

http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/feature/496185.rdf

Most of the links referred in the response are resolvable (FeatureClass, Admin1)

Example: Get the FeatureClass School description

http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/featureclass/School

The endpoint returns the following response in Turtle format

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/featureclass/Sch
ool>
        a       <http://www.opengis.net/ont/gazetteer#LocationType> ;
        <http://purl.org/dc/terms/description>
                "Building or group of buildings used as an institution
for study, teaching, and learning (academy, college, high school,
university)." .
```

Admin2 level Harrison county, KY

http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/admin2/KY.097

The endpoint returns the following response in Turtle format

```
<http://ows.usersmarts.com/ldapp/ows11/gazetteers/gnis/admin2/KY.097>
        a       <http://www.opengis.net/ont/gazetteer#Location> ,
<http://www.opengis.net/ont/gazetteer#County> ;
        <http://www.w3.org/2000/01/rdf-schema#label>
                "Harrison" ;
        <http://www.opengis.net/ont/gazetteer#locationType>

<http://www.opengis.net/ont/gazetteer#AdministrativeAreaLevel2> ;
        <http://www.opengis.net/testbed11/ont/gnis#countyId>
                "097" .
```

13.3.3.2 Version 2: WFS-G semantic mapping of GNIS (broken at present).

The second version is based on WFS-G provided by Compusult. It uses a semantic mapping on top of the WFS. GeoSPARQL queries are converted to OGC Query and the GML response is converted back to RDF. For the OWS 11 demonstration, we made most of the URL of the resources in GNIS resolvable and accessible through REST API. The data can be returned in RDF, TTL, N2, NT, or JSON-LD format. The SPARQL endpoint provides a HTML-based client with syntax validation and allows to visualize the results in raw format, table or visual view when the data structure format. The mapping is pretty limited due to technical challenge related to XLink (for example LocationTypeidenfitifer requires to traverse the link for each solution which makes the query engine highly inefficient).

The SPARQL Endpoint is located at:

http://ows.usersmarts.com/ldapp/ows11/gazetteers/usgs/sparql

13.4 Semantic Social Media Scraper Service

To demonstrate the Linked Data wrapper approach for integrating Social Media information and APIs, a RESTful Linked Data Scraper Service was implemented by Image Matters. The service was proposed as an alternative to the Sensor Observation Service (SOS). It demonstrated the use of RESTful API, Linked Data Platform (LDP) specification and SocialML ontologies. Three LD Scraper types were implemented during the testbed: Twitter, YouTube and Flickr.

With the large variety of social media APIs producing feeds of social objects and activities, it is difficult to integrate information into a coherent framework that supports analysis that leads to informed decisions. Using a common core vocabulary to describe social items and activities, we can extend and accommodate the specificities of each APIs without breaking the coherency of the representation of the social information generated by these sites. The Linked Data representation of the social information provides a consistent, unified representation and access using Linked Data standards such as RDF, OWL, SPARQL and OGC GeoSPARQL. In addition to access, the use of ontologies enables powerful reasoning capabilities that relieve the cognitive burden on users trying to use social media as actionable information for decision support.

The integration of social medias APIs was done using a three layer approach:

The first layer is composed of the Social Media APIs producing syntactic based social information encoded most of the time in JSON or XML forms. This layer is bridged to a semantic layer representation using RDF scrapers plugins that convert data from a given API to a linked data representation using the SocialML vocabularies and its extensions. By using a registry of RDF Scraper plugins, native APIs now can be accessed using a unified approach and producing a unified representation of the social information. This Linked Data information can be aggregated and stored in a RDF store that could be accessed either through a GeoSPARQL endpoint or using Linked Data API that provided a RESTful access to Social Items and Activities resources. The GeoSPARQL endpoint could be used to perform different analytics using computational agents (OGC Web Processing Services) or human based consumption. In addition to analytics, the Social information can be easily linked to other relevant information such as features to augment the value of existing information. The functional architecture of the service is illustrated in Figure 14.

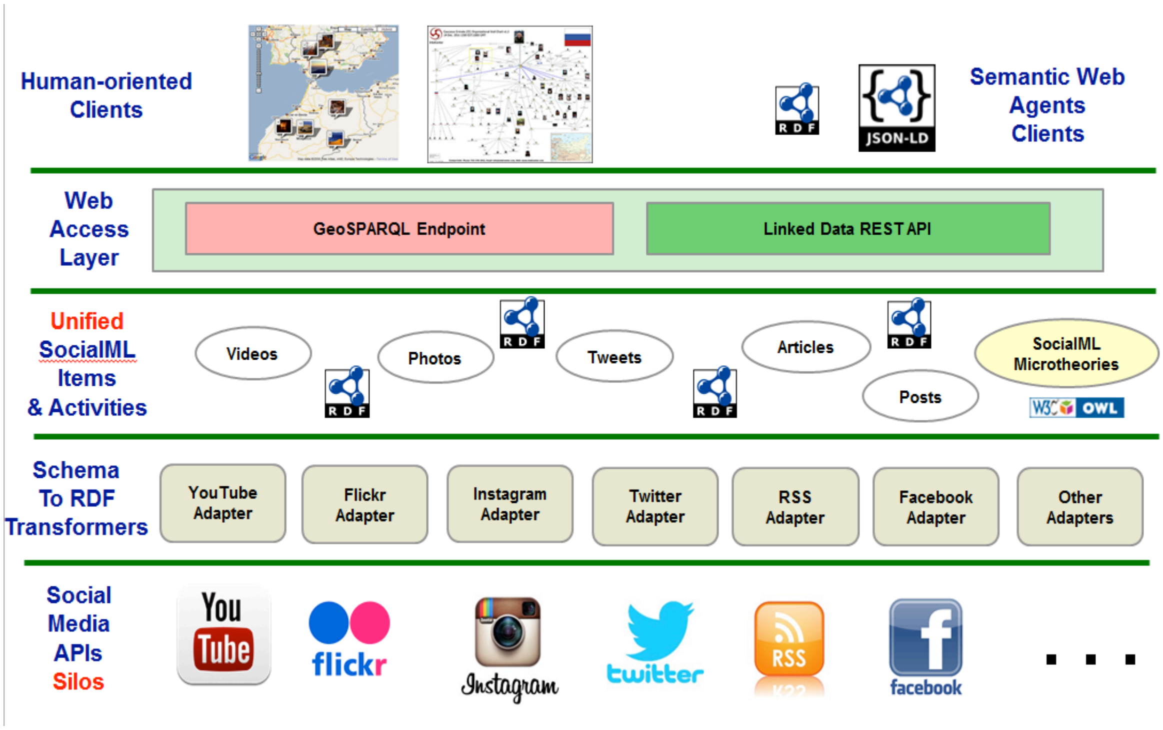


Figure 14 Functional architecture of Semantic Social Media Scraper Service

More details about the service REST API can be found in the Testbed 11 Engineering Report ER 15-057 - Incorporating Social Media in Emergency Response Engineering Report.

14 Recommendations

14.1.1 Geospatial Ontologies

The core geospatial ontologies are the result of eight years of Research and Development at Image Matters in the domain of geospatial semantics. Image Matters decided to release these ontologies for the broad community seeking to facilitate semantic interoperability between systems using geospatial information. The logical next steps are a critical review of the provided ontologies and the application of necessary refinements and extensions to achieve a comprehensive set of base ontologies. To favor the adoption of the geospatial ontologies, improve the robustness and completeness of the micro-theories, we suggest for the next testbed exercising the ontologies by converting existing geospatial data into geospatial knowledge, creating catalogs for unit of measures, CRSs and feature types (for gazetteers). The ontologies can also be exercised by creating ontologies for more specialized domain that leverage the core geospatial ontologies, as it was done for this testbed for E&DM. The result of these activities should lead to identifying best practices and provides a rich set of examples how geospatial ontologies could be used for different vertical communities and be queried using open linked data standards and protocols (Linked Data Platform, GeoSPARQL). We propose to identify in the next testbed the microtheories that bring the most values for the community and coordinate with the work of Spatial Data on the Web WG to promote them as best usage practices.

14.1.2 GeoSPARQL 2.0 amd GeoSPARQL extenstions

Based on feedback from the OGC Geosemantic WG and lessons learned from Testbed 11, there is a need to modularize and simplify but also extend GeoSPARQL specification. The Testbed 10 and 11 geospatial ontologies address many of these aspects (for example modularization of spatial relations), and could be used as a starting point. The GeoSPARQL extensions ontology in the Symbology Mediation provides powerful mechanism to expand the usage and capabilities of GeoSPARQL. The next testbed should demonstrate the feasibility and robustness of the approach by implementing the specifications.

14.2 REST enablement OGC services

During this testbed, we found that the use of RESTful API for services was easier to semantically enable as Linked Data best practices built on RESTful foundations. We demonstrated that it is feasible to leverage existing infrastructure such as spatial RDBMS to serve geospatial information as Linked Data without loss of performance and using RESTful API and GeoSPARQL query language as a unifying query language. The use of Resource Oriented Architecture (ROA) best practices and Linked Data best practices helps to enforce correct design of RESTful service APIs. The Linked Data Platform API provides also a solid framework to perform CRUD operations using Linked Data representation and should be investigated further in future testbeds. The use of resolvable URIs for Linked Data and ontologies should be used as best practices in order to facilitate linking and integration of different sources as demonstrated for the NHD and GNIS integration. JSON-LD also provides a promising solution to bridge Linked Data with mainstream web development requiring JSON-LD. More testing and deployment using JSON-LD needs to be investigated in future testbed,

We recommend prioritizing the semantic enablement of services based on maximizing the value proposition they can bring to the communities of interest. We propose investigating the following services as candidates for semantical-enablement in the next testbed:

- Semantic Registry (or Catalog) enables better discovery and access of services and data using semantic representation of datasets and services. DCAT is a good starting point. Image Matters is currently working on an extension of DCAT to describe services and geospatial datasets (called GeoDCAT).

- Semantic Gazetteers: Defining a formal ontology for describing Location and LocationType as well as Linked Data REST API to access Location information has a wide range of application cross-cutting many communities of interests.
- Semantic WFS: Leveraging Linked Data API to share Feature semantically can facilitate the integration of geospatial information using standard RESTful API and Linked Data representation.

15 Revision history

Date	Release	Editor	Primary clauses modified	Description
06/10/2015	0.1	Stephane Fellah	Initial Outline	
07/01/2015	0.2	Stephane Fellah	Secion 9	Relevant ontology section
07/07/2015	0.3	Stephane Fellah	Section 6 and 7	Semantic Web and Linked Data sections
07/19/2015	0.4	Stephane Fellah	All	
07/20/2015	1.0	Stephane Fellah	All	Completion of missing sections
09/22/2015		Carl Reed	Various	Final edits in preparation for publication

Bibliography

[1] Guidelines for Successful OGC Interface Standards, OGC document 00-014r1

[2] W3C Best Practice Recipes for Publishing RDF Vocabularies.

[3] Sauermann, Leo, Richard Cyganiak, and Max Völkel. "Cool URIs for the semantic web." (2011).

[4] Berrueta, Diego, Jon Phipps, Alistair Miles, Thomas Baker, and Ralph Swick. "Best practice recipes for publishing RDF vocabularies." Working draft, W3C(2008).

[5] Deirdre Lee; Bernadette Farias Lóscio; Phil Archer. Data on the Web Best Practices Use Cases & Requirements. Note. URL: http://www.w3.org/TR/dwbp-ucr/

[6] Fadi Maali; John Erickson. *Data Catalog Vocabulary (DCAT)*. 16 January 2014. W3C Recommendation. URL: http://www.w3.org/TR/vocab-dcat/

[7] Timothy Lebo; Satya Sahoo; Deborah McGuinness. *PROV-O: The PROV Ontology*. 30 April 2013. W3C Recommendation. URL: http://www.w3.org/TR/prov-o/

[8] Bernadette Hyland; Ghislain Auguste Atemezing; Boris Villazón-Terrazas. *Best Practices for Publishing Linked Data*. 9 January 2014. W3C Note. URL: http://www.w3.org/TR/ld-bp/

[9] W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. 11 December 2012. W3C Recommendation. URL: http://www.w3.org/TR/owl2-overview/

[10] Dan Brickley; Ramanathan Guha. RDF Schema 1.1. 25 February 2014. W3C Recommendation. URL: http://www.w3.org/TR/rdf-schema/

[11] RDF 1.1 Turtle: Terse RDF Triple Language. *W3C Recommendation, 25 February 2014. The latest edition is available at* http://www.w3.org/TR/turtle/

[12] SKOS Reference, W3C Recommendation, 18 August 2009. Latest version available at http://www.w3.org/TR/skos-reference .

[13] Bernadette Farias Lóscio, Caroline Burle, Newton Calegari, " Data on the Web Best Practices" W3C Editor's Draft 18 May 2015