Publishing and Consuming Government Linked Data on the Semantic Web

Ghislain Auguste Atemezing

A doctoral dissertation submitted to:

TELECOM ParisTech

in partial fulfillment of the requirements for the degree of:

DOCTOR (PhD)

Specialty : Semantic Web

Jury :

Reviewers:
Examiners:
Supervisors:

To all those who helped me
to make this dream coming true.
"Things don't have to change
the world to be important".
-Steve Jobs -

Acknowledgements

This thesis is the result of part of my work carried out in the context of two Projects: the Datalift Project (ANR-10-CORD-009), and the Apps4Europe Project. There are many people I want to thank since they kindly supported me in so many different ways for the successful completion of this thesis.
I am indebted to my thesis advisor Dr. Raphaël Troncy for giving me the opportunity for a PhD. at EURECOM / Telecom ParisTech. Throughout my PhD he provided helpful ideas and encouraging support in this exciting domain of Web Semantic.
I would like to thank my committee members, the reviewers Prof. XXX and Dr. XXXX, and furthermore the examiners Dr. XX and Dr. XXX for their precious time, shared positive insight and guidance.
I would like to express my deepest appreciation for all the colleagues and partners of Datalift project for all the exchanges during the duration of the project. I would like to thank specially Bernard, Pierre-Yves, Nathalie, Laurent. It was a pleasure to work and exchange with them.
My sincere thanks are due to all the members of the W3C Government Linked Data Working Group, specially to Bernadette Hyland, Boris Villazón-Terrazas, Richard Cyganiak, Dave Reynolds and Phil Archer. We had many intensive moments of exchange and collaborations.
I would like also to show my sincerest gratitude to all my mentors at OEG/UPM (Madrid) for providing me first skills on ontology engineering: Pr. Asuncion Gómez and Dr. Oscar Corcho. Their enthusiasm and dedication to research on this field were truly inspiring.
My warmest thanks to my colleagues who supported me during my Ph.D. Precisely, I would like to thank Houda, Giuseppe, José Luis, Vuk and Ahmad. Also, I thank all those working at EURECOM, they made my stay at EURECOM very pleasant.
I would like to express my deepest thanks to my wife, Verónica Alvarez, who gave me all her patience and comprehension since the beginning of this adventure. I owe my family my profound gratitude because they always supported and believed in me: specially my parents Genevieve Djifack and Prosper Tabondjou; and all my brothers and sisters: Stephanie, Judith, Arlette, Mathias, Alvine, Edwige, Yannick.
Lastly, special thanks to my friends for their unwavering friendship, moral and infinite support.

Abstract

The French national mapping agency (IGN) produces different but complementary geographic vector reference databases delivered in traditional GIS formats. However, linked data users have different expectations and habits, such as the need to browse an entire data catalog in RDF using the "follow-your-nose" navigation capacity from one graph to another. Besides, traditional GIS data formats are not interoperable with RDF. Yet, all these geographic datasets could be used with benefits on the Web of data, either with direct georeferencing through geographic primitives, or indirect one through postal addresses. We have contributed to the georeferencing of datasets published on the Web of data by providing such resources for the French territory. Firstly, we propose two vocabularies designed for representing structured geometries defined with coordinates expressed in any Coordinates Reference System (CRS). Secondly, we reuse these vocabularies and the CRSs' dataset to publish a reference dataset on administrative units that can also be reused for indirect georeferencing purposes. Finally, we also propose two vocabularies for describing geographic feature types. In addition to these resources, we also present a comprehensive workflow for easily publishing geographic data on the Web of data.

Moreover, it is widely accepted that by controlling metadata, it is easier to publish high quality data on the web. Metadata, in the context of Linked Data, refers to vocabularies and ontologies used for describing data. With more and more data published on the web, the need for reusing controlled taxonomies and vocabularies is becoming more and more a necessity. Catalogues of vocabularies are generally a starting point to search for vocabularies based on search terms. Some recent studies recommend that it is better to reuse terms from "popular" vocabularies [1]. However, there is not yet an agreement on what makes a popular vocabulary since it depends on diverse criteria such as the number of properties, the number of datasets using part or the whole vocabulary, etc. We propose a method for ranking vocabularies based on an information content metric which combines three features: (i) the datasets using the vocabulary, (ii) the outlinks from the vocabulary and (iii) the inlinks to the vocabulary. We applied this method to 366 vocabularies described in the LOV catalogue. The results are then compared with other catalogues which provide alternative rankings.

Finally, datasets published in the LOD cloud can often be accessed by different means such as API access, bulk download or as linked data fragments, and most of the time, a SPARQL endpoint is also provided. While the LOD cloud keeps growing, having a quick glimpse of those datasets is getting harder and there is a need to develop new methods enabling to detect automatically what an arbitrary dataset is about and to recommend visualizations for data samples. We consider that "a visualization is worth a million triples", and we propose a novel approach that mines the content of datasets and automatically generates visualizations. Our approach is directly based on the usage of SPARQL queries that will detect the

important categories of a dataset and that will specifically consider the properties used by the objects which have been interlinked via `owl:sameAs` links. We then propose to associate type of visualization for those categories. We have implemented this approach into a so-called Linked Data VIzualization Wizard (LDVizWiz).

Contents

List of Figures

List of Tables

Listings

List of Publications

Book

- Atemezing, Ghislain Auguste and Troncy, Raphaël: Multimedia metadata. In "Encyclopedia of Social Network Analysis and Mining", Springer Verlag, 2014, ISBN: 978-1461461692

Journal

1. Atemezing, Ghislain and Corcho, Oscar and Garijo, Daniel and Mora, José and Poveda-Villalón, María and Rozas, Pablo and Vila-Suero, Daniel and Villazón-Terrazas, Boris: **Transforming meteorological data into linked data**. In Semantic Web journal, Special Issue on Linked Dataset descriptions, 2012 (to appear). IOS Press, ISSN: 1570-0844.

2. Suárez-Figueroa, Mari Carmen; Atemezing, Ghislain Auguste; Corcho, Oscar : **The landscape of multimedia ontologies in the last decade** in Multimedia Tools and Applications, Vol 55, NÂ°3, December 2011.

Conferences and Workshops

1. Atemezing, Ghislain Auguste; Troncy, Raphaël: **Towards a linked-data based visualization wizard**. In COLD 2014, 5th International Workshop on Consuming Linked Data, 19 October 2014, Riva del Garda, Italy .

2. Governatori, Guido and Lam, Ho-Pun and Rotolo, Antonino and Villata, Serena and Atemezing, Ghislain and Gandon, Fabien: **Checking licenses compatibility between vocabularies and data**. In COLD 2014, 5th International Workshop on Consuming Linked Data, 19 October 2014, Riva del Garda, Italy.

3. Governatori, Guido and Lam, Ho-Pun and Rotolo, Antonino and Villata, Serena and Atemezing, Ghislain and Gandon, Fabien: **LIVE: a Tool for Checking Licenses Compatibility between Vocabularies and Data**. In DEMO Session at ISWC'2014, 19 October 2014, Riva del Garda, Italy.

4. Atemezing, Ghislain Auguste and Abadie, Nathalie and Troncy, Raphaël and Bucher, Bénédicte, **Publishing reference geodata on the web: Opportunities and challenges for IGN France** . In TERRA COGNITA 2014, 6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web, October 19-23, 2014, Riva del Garda, Italy.

5. Atemezing, Ghislain Auguste; Troncy, Raphaël: **Information content based ranking metric for linked open vocabularies**. In SEMANTICS 2014, 10th International Conference on Semantic Systems Proceedings, 4-5 September 2014, Leipzig, Germany.

6. Assaf, Ahmad and Atemezing, Ghislain Auguste and Troncy, Raphaël and Cabrio, Elena : **What are the important properties of an entity? Comparing users and knowledge graph point of view**. In ESWC 2014, 11th Extended Semantic Web Conference, Heraklion, Crete.

7. Troncy, Raphaël; Atemezing, Ghislain Auguste; Abadie, Nathalie; Lam, Cao-Vien. **Modeling geometry and reference systems on the web of data**. In W3C 2014, W3C Workshop on Linking Geospatial Data, March 5-6, 2014, London, UK.

8. Atemezing, Ghislain Auguste; Vatant, Bernard; Troncy, Raphaël ; Vanderbussche, Pierre-Yves. **Harmonizing services for LOD vocabularies: a case study**. In WASABI 2013, Workshop on Semantic Web Enterprise Adoption and Best Practice, 22 October, 2013, Sydney, Australia.

9. Feliachi, Abdelfettah; Abadie, Nathalie ; Fayćal, Hamdi; Atemezing, Ghislain Auguste. **Interlinking and visualizing linked open data with geospatial reference data**. Poster in Ontology Matching Workshop, 22-23 October 2013, Sydney, Australia.

10. Atemezing, Ghislain Auguste; Gandon, Fabien; Kepeklian, Gabriel; Scharffe, François; Troncy, Raphaël; Villata, Serena: **When publishing linked data requires more than just using a tool**. In W3C 2013, Workshop on Open Data on the Web, April 23-24, 2013, London, UK.

11. Atemezing, Ghislain Auguste; Troncy, Raphaël: **Towards interoperable visualization applications over linked data**. In EDF 2013, 2nd European Data Forum, April 9-10, 2013, Dublin, Ireland.

12. Scharffe, François; Atemezing, Ghislain; Troncy, Raphaël; Gandon, Fabien; Villata, Serena; Bucher, Bénédicte; Hamdi, Fayçal; Bihanic, Laurent; Képéklian, Gabriel; Cotton, Franck; Euzenat, Jérôme; Fan, Zhengjie; Vandenbussche, Pierre-Yves; Vatant, Bernard: **Enabling linked-data publication with the datalift platform** in AAAI 2012, 26th Conference on Artificial Intelligence, W10:Semantic Cities, July 22-26, 2012, Toronto, Canada.

13. Atemezing, Ghislain; Troncy, Raphaël : **Vers une meilleure interopérabilité des données géographiques françaises sur le Web de données** in IC 2012, 23émes Journées Francophones d'Ingénierie des Connaissances, June 25-29, 2012, Paris, France.

14. Khrouf, Houda; Atemezing, Ghislain; Rizzo, Giuseppe; Troncy, Raphaël; Steiner, Thomas: **Aggregating social media for enhancing conference experiences** in RAMSS 2012, 1st International Workshop on Real-Time Analysis and Mining of Social Streams, June 4, 2012, Dublin, Ireland.

15. Khrouf, Houda; Atemezing, Ghislain; Steiner, Thomas; Rizzo, Giuseppe; Troncy, Raphaël: **Confomaton: A conference enhancer with social media from the cloud** in ESWC 2012, 9th Extended Semantic Web Conference, May 27-31, 2012, Heraklion, Crete.

W3C Documents

1. Hyland, Bernadette ; Atemezing, Ghislain ; Villazón-Terrazas, Boris: **Best Practices for Publishing Linked Data**. W3C Working Group Note published on January 9, 2014. Url: `http://www.w3.org/TR/ld-bp/`

2. Hyland, Bernadette ; Atemezing, Ghislain ; Pendleton, Michael ; Srivastava, Biplav: **Linked Data Glossary**. W3C Working Group Note published on June 27, 2013. Url: `www.w3.org/TR/ld-glossary/`

Acronyms

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first appears in the text.

apps4x	Apps for X Co-creation Event Vocabulary
CRS	Coordinate Reference System
DCAT	Data Catalog Vocabulary
DOM	Data Object Model
DSRM	Data State Reference Model
DVIA	The Data VIsualization Application Vocabulary
French Linked Open Data	FrLOD
LD	Linked Data
LDA	Linked Data API
LDVM	Linked Data visualization Model
LOD	Linked Open Data
LOV	Linked Open Vocabulary
GDAL	Geospatial Data Abstraction Library
GLD	Government Linked Data
GI	Geographic Information
GIS	Geographic Information System
GKP	Google Knowledge Panel
GPS	Global Positioning System
odapps	Open Data Applications Vocabulary
OGC	Open Geospatial Consortium
Partitioned Information Content	PIC
Information Content	IC
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SDI	Spatial Data Information
SFA	Simple Features Access
SPARQL	SPARQL Protocol and RDF Query Language
UI	User Interface
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium

Chapter 1

Introduction

"The Web as I envisaged it, we have not seen it yet. The future is still so much bigger than the past."
Tim Berners-Lee

1.1 Context

The Web is currently in a transition phase. After having been accessible on personal computers, it is now quickly moving to more and more ubiquity and entering in every part and moment of our lives. New devices and new ways to use them are being created. The ubiquity of the Web also creates an unseen abundance of information. Data is flowing onto the Web, created by users, generated by sensors, and stored in ever growing data farms. Geographic data is widely present on the web as they are used for location of Point of Interest. At the same time, many organizations are moving from legacy data stored in their databases to structured data on the web. Structured data is already present in the many databases, metadata attached to medias, and in the millions of spreadsheets created everyday across the world.
The Web of Linked Data, unlike the web of hypertext, is constructed with documents on the web data with links between arbitrary things described by RDF. Hence, the URIs identify any kind of object or concept [5]. It is continuously evolving, started in 2007 with a dozen of datasets (cf.Figure1.1) to a large data space with thousands of datasets in different topics. From 2011 (See Figure 1.2)[6] to 2014, there has been a significant growth of nearly 271% of datasets depicted in the LOD cloud [7]. The new version altogether contains 570 linked datasets which are connected by 2909 linksets, as depicted in Figure 1.3[1]. In order to enable Linked Data applications to discover datasets as well as to ease the integration of data from multiple sources, Linked Data publishers should comply with a set of best practices for publishing datasets on the web [8]:

- **Data selection:** The dataset should be selected based on its potential relevance to be reused in an open format accessible somewhere on the Web.
- **Vocabulary Usage:** The best practices advise publishers to use terms from widely-used vocabularies in order to ease the interpretation of their data. If data providers use their own vocabularies, the terms of such proprietary vocabularies.

[1] A more web friendly version can be accessed at `http://data.dws.informatik.uni-mannheim.de/lodcloud/2014/`

- **Linking:** By setting RDF links, data providers connect their datasets into a single global data graph which can be navigated by applications and enables the discovery of additional data by following RDF links.
- **Dereferencable URIs:** By using HTTP URIs as identifiers for each resource, agents can easily look-up at the resources and "dereference" a URI in order to have access to the full representation identified by that URI. This helps building a network of URIs on the Web and navigating through different graphs.
- **Metadata Provision and machine access to data:** Provide various ways for search engines and other automated processes to access data using standard Web mechanisms.

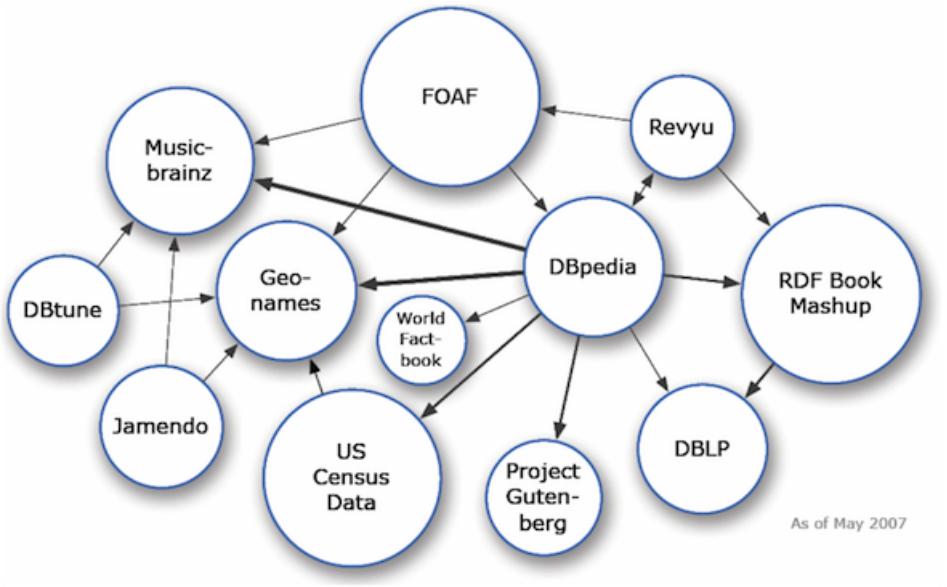


Figure 1.1: LOD cloud as of May, 2007

The Web is currently in a transition phase. After having been accessible on personal computers, it is now quickly moving to more and more ubiquity and entering in every part and moment of our lives. New devices and new ways to use them are being created. The ubiquity of the Web also creates an unseen abundance of information. Data is flowing onto the Web, created by users, generated by sensors, and stored in ever growing data farms. Geographic data is widely present on the web as they are used for location of Point of Interest. But this data still lack of interoperability for a better integration due to these three main factors:

- Vendor specific geometry support, such as Google Maps API, Yahoo Geo Technologies, etc.
- Different vocabularies, such as W3C Basic Geo, NeoGeo, GeoSPARQL, GML XMLLiteral and vendor-specific.

- Different spatial reference systems, such as Lambert93, WGS84, British National Grid, etc.

At the same time, many organizations are moving from legacy data stored in their databases to structured data on the web. Structured data is already present in many databases, metadata attached to medias, and in the millions of spreadsheets created everyday across the world.
One of the key benefit of Linked Data is the use of RDF model to manipulate data on the Web, interconnect it with other data and consume it in a variety of applications. In the process of getting those datasets effectively published, there are some barriers that prevent publishers to embrace the movement, such as the following:

- RDF and it's different serializations is difficult to understand and use in practice compare to CSV or JSON.
- Having a dataset, choosing a suitable vocabulary to model the data is a big challenge.
- There are not much easy tools to guide the publishers in their process of publishing their dataset without be specialist in the different technologies: such as SPARQL, server configurations for dereferencing URIs, etc.
- The tools for converting data into RDF are either more domain-specific, or difficult to configure for non experts.

Nevertheless, the resulting "Web of data" has started being populated in different domains, particularly with geospatial data, as proved by the efforts in [9, 10, 11, 12]. Those efforts and initiatives follow the vision of the *Semantic Geospatial Web* promotes by Max Egenhofer in [13] challenging GIS researchers to contribute to the Semantic Web effort by creating geospatial ontologies, query languages and processing techniques adapted to geospatial information on the Web.
At the same time, the recent emergence of Linked Data radically changes the way structured data is being considered. By giving standard formats for the publication and interconnection of structured data, linked data transforms the Web into a giant database. While making data available on the Web, we need to build meaningful applications to show the value of all the huge data so that users could easily explore it, and derive new insights for it. As many information visualization tools are already present in InfoVis community[2], their easy adoption and usage for displaying structured data raise new challenges. Those challenges are two-folds:

- How to specify and define semantic web applications in terms of tools, widgets that can easily visualize RDF datasets?
- How to mine efficiently heterogeneous structured data to derive patterns for automatically recommend the adequate visualization tool to help users building innovative applications in an affordable time.

[2] `http://en.wikipedia.org/wiki/Information_visualization`

- How could we bridge the gap between traditional infoVis tools and Semantic Web technologies to built easily applications on top of datasets published as L(O)D?
- How to represent and share visualizations built with datasets already present in different Open data portals, such as `data.gouv.fr` and `data.gov.uk`.

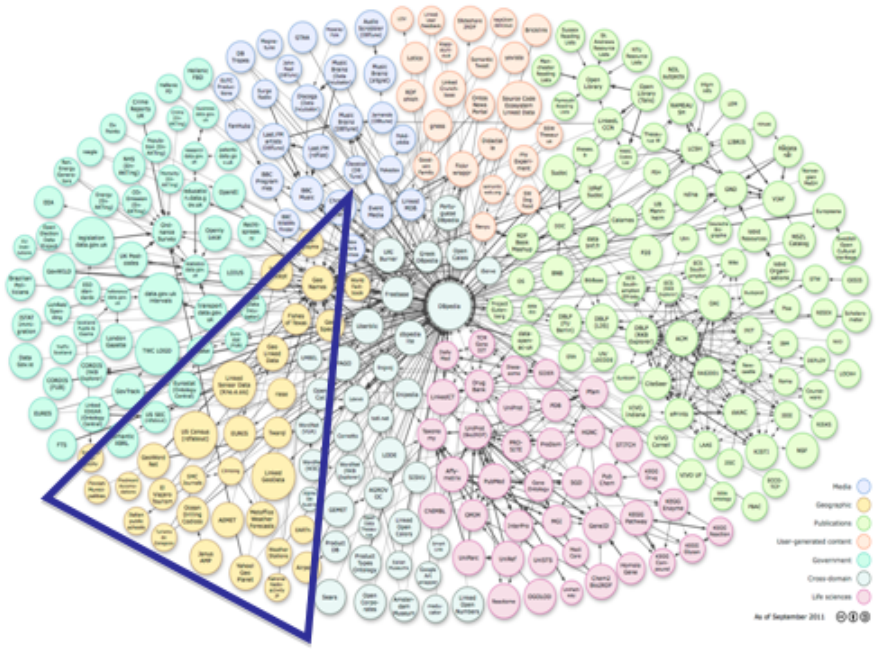


Figure 1.2: Linking Open Data cloud diagram 2011, by Anja Jentzsch and Richard Cyganiak. `http://lod-cloud.net/`

1.2 Research Questions

The ubiquity of the Web is creating an unseen abundance of information. Data is flowing onto the Web, created by users, generated by sensors, and stored in ever growing data farms. Geographic data is widely present on the web as they are used for location of Point of Interest. At the same time, many organizations are moving from legacy data stored in their databases to structured data on the Web. Structured data is already present in many databases, metadata attached to medias, and in the millions of spreadsheets created everyday across the world. Many Linked Open Datasets have geospatial components, but still not having a common ways to describe features, spatial objects or geometries. Let us take the following three use-cases to express how challenging is to integrate geographic data from different datasets to obtain relevant answers:

`UC1: What DBpedia Historic Buildings are within walking distance`

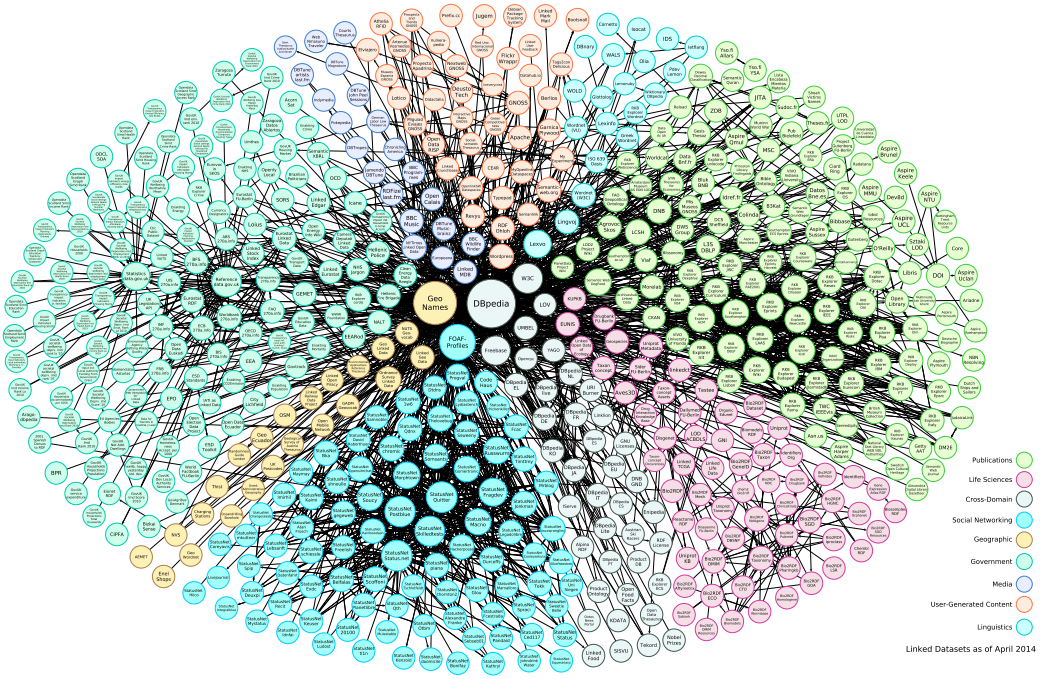


Figure 1.3: Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. `http://lod-cloud.net/`

```
     from my current location ?
UC2: What departments are located inside the bounding box
     composed of Eurecom location and the Eiffel Tower?
UC3: Give me the centroids of the administrative units
     in France in Lambert93?
```

The aforementioned use-cases take into account "Concepts" (e.g: Historic Building, Department) that are defined differently depending on the provider of the dataset (e.g: DBpedia, OpenStreetMap, IGN). Besides, the aforementioned use-cases implicitly make use of some specific topological functions widely used in the GIS applications, such as "within" , "inside", "bounding box". For example, Use Case 2 also mentioned Lambert93, a specific projection of data in France. Our aim is to contribute to the actual efforts in representing geographic objects to leverage the barrier of integration of geospatial data both by the pubishers and the lay users consuming the data.
Our main concern is to tackle the problematic within the workflow of publication in two directions, more likely to happen at the beginning and the end:

- (i) Geographic Information on the Web of data: as an application of the lifecycle of publishing geodata.
- (ii) Visualization tools for building innovative applications consuming struc-

tured data: as for leveraging the process of creating applications on-top of semantic data to highlight some relevant knowledge to the users.

In [14], the authors mentioned many research topics in the area of linked geospatial data. In this thesis, we try to find answers to the following challenges:

1. *Vocabularies:* How do we model geospatial information on the Web? How do we assess geospatial ontologies? How to serialize complex geometry?
2. *Query languages:* How do we write efficient queries that target geospatial Web? How do store and index geodata in RDF ?
3. *Datasets:* How do we extract and convert geodata to expose on the Web? What are the best practices for representing complex geometries on the Web? How can we integrate fully compatibility of CRSs on datasets?
4. *Publication:* How can we develop scalable frameworks for covering the workflow of publishing geodata? What are the appropriate triple stores for handling geodata? What are the metrics to use for interconnecting different geodata resources on the Web?
5. *Applications and user interfaces:* How do we generate interesting visualizations of linked geospatial data? What are appropriate high-level APIs that ease the development of user interfaces for geospatial data? Can we rely on existing map platforms such as Google Maps, Bing Maps or OpenSteetMap?

In this thesis we mainly focus on the first, third, fourth and fifth of the above topics, and acknowledge existing efforts on the rest of the topic. We tackle the issues and challenges both from publishers and users point of view. The publishers need pragmatic solutions that help to choose a vocabulary, find a tool to convert ShapeFiles according to well-known vocabularies, then generate and publish the data following some best practices.

1.3 Contributions

In this section, we provide the main contributions organized in three main sections of our thesis: model and publication of geospatial data, visualizations of data and applications on the Web and contribution to standards.

1.3.1 Modeling Geographic Information in LOD

The need for geolocation is crucial for many applications for both human and software agents. More and more data is opened and interlinked using Linked Data principles, and it is worth modeling geographic data efficiently by reusing as much as possible from existing ontologies or vocabularies that describe both the geospatial features and their shapes. In the first part of our work, we survey different modeling approaches used by the Geographic Information System (GIS) and the Linked

Open Data (LOD) communities. Our aim is to contribute to the actual efforts in representing geographic objects with attributes such as location, points of interest (POI), and addresses on the web of data. We focus on the French territory and we provide examples of representative vocabularies that can be used for describing geographic objects. We propose some alignments between various vocabularies (DBpedia, GeoNames, Schema.org, LinkedGeoData, Foursquare, etc.) in order to enable interoperability while interconnecting French geodata with other datasets.
Regarding this aspect of our research, we have achieved the following tasks:

- We have proposed an ontology describing features and point of interest for the French territory, by reusing existing taxonomy (GeOnto) and aligning it to other related vocabularies of the domain.
- We studied how to extend the existing vocabularies for geographic domain to take into account efficient modeling for complex geometries. By doing so, tackle the complex geometry representation issues in the Web of Data, describing the state of implementations of geo-spatial functions in triple stores and comparing them to the new GeoSPARQL standard. We finally make some recommendations and advocate for the reuse of more structured vocabulary for publishing topographic entities to better address the IGN-France requirements.
- We have made a comparative study of the triple stores, comparing their capability to store spatial information and their implementation of topological functions with respect to the ones existing in OGC[3] standards.
- We have designed and implemented vocabularies for describing complex geometries with different coordinate systems, with direct application to the French administrative units.

1.3.2 Visualization Tools in Linked Government Data

We first review the numerous applications that have been developed on top of datasets that have been opened by governments (UK, USA, France) and local authorities. We have then derived and proposed some use cases (8 UCs) that can be developed to consume data from the different main providers in the French level: INSEE, DILA, IGN, FING, etc. We mention that the most interesting UCs are the ones which show the added value of having interconnected datasets. These UCs, developed and deployed, can be useful to show the benefits of Linked Data in a variety of domains such as Education, Tourism, Cultural Heritage, Civil administrations, Judicial Court, Medicine, etc.
Regarding tools used for visualization, we have divided them in two categories, providing for each of them relevant examples: (i)-tools that operate over RDF data, (ii) and tools that operate over other structured format. We then provide some

[3] `http://www.opengeospatial.org/`

basic criteria for assessing a given visualization tool, with some weight attached to each of the criterion.

1.3.3 Contributions on visualizations

Concerning our contributions on visualizations, we have contributed by:

- Building an application of the French first round elections using data from the `data.gouv.fr` and other public institutions.
- Implementing a generic tool for exploring geodata on a map.
- Contributing in the creation of a French LOD Cloud by publishing many datasets as LOD covering the French territory.
- Implementing an application consuming geodata and statistics combining multiple datasets in education from `http://data.gouv.fr` portal.
- Building an application for conference events (confomaton) with their associated media reconciled from many social platform (instagram, twitter, etc.).
- Building a vocabulary for structuring applications on the Web of data. A plugin for annotating applications developed for contests is implemented to ease the use the generation of structured data using the vocabulary.
- Implementing a wizard that analyses an RDF dataset and recommend visualization based on predefined categories, using generic SPARQL queries for easing the exploration of datasets published as LOD.

1.3.4 Contributions to Standards

We contributed to the W3C Government Linked Data Working Group (GLD WG)[4] activity from July, 2011 until December, 2013. The objective of the Working Group was to *"provide standards and other information which help governments around the world publish their data as effective and usable Linked Data using Semantic Web technologies"*.
The group had three main task forces:

- **Task Force #1** aims to create a linked data community directory[5] and to maintain it on-line about deployments, vendors, contractors, end-user applications. In this work, we contribute to define the requirements and providing data for the French organizations in the directory.
- **Task Force #2** aims at providing **"Best Practices"** for Publishing Linked Data by producing recommendations regarding vocabulary selection, URI construction, Linked Data Cookbook, versioning, stability and provenance. Here,

[4] http://www.w3.org/2011/gld/
[5] http://dir.w3.org

we have prepared a check- list to help government to select and re-use vocabularies in their project. We have also proposed our vision of the Linked Open Data Life cycle, best practices for creating URIs. We were editor for the Linked Data Glossary [82] published as Note document. Apart from contributing in many sections of the document "Best Practices for Publishing Linked Data" [3].

- **Task Force #3** goal was to provide relevant vocabularies to be used by governments or local authorities in their process of exposing their data. We have participated actively in the discussions on the different vocabularies published as recommendations by the W3C such as Data Cube [15], ORG vocabulary [16] and DCAT [65] vocabulary.

Regarding the use of standard vocabularies we have contributed on :

- Proposing a method to harmonize prefixes on the web of data with two services: Linked Open Vocabularies (LOV)[6] and prefix.cc[7]. The former is currently a maintained hub of curated vocabularies on the Web, while the latter is a focal point for developer to register and look-up prefixes for their resources or ontologies. The approach proposed can be extended to any catalogue of vocabulary as long as the vocabularies fulfill the requirements to be inserted into LOV catalogue.
- Designing and implementing a new method for ranking vocabularies based on the computation of Information Content (IC) and Partitioned Information Content (PIC) metrics.
- We have developed a tool than answer in real-time whether different licenses present in the dataset and vocabularies are either compatible or not.

1.4 Thesis Outline

The work presented within this thesis is composed of three major parts. The remainder of the thesis proceeds as follows:

- In the first part of this thesis, we focus on the various models/vocabs for representing geography/geometry. We survey the state of the triple stores and describe particular problems (coordinate systems, etc.) and highlight our contributions: new vocabularies, an online converter between CRS, etc. We also describe how geography database can then be converted into RDF using the Datalift process. We then show how those datasets can be interlinked (possibly trying different instance matching tools) and it will conclude with a thorough analysis of those alignments in the case of the IGN-France datasets. More specifically:

[6] `http://lov.okfn.org/dataset/lov/`
[7] `http://prefix.cc`

  - **Chapter 2** describes the current limitations of geodata on the Web and the different vocabularies we propose for geometries, coordinate reference systems and feature types. We also propose some best practices to publish geodata on the web.
  - **Chapter 3** focuses on tools for publishing and querying geodata, their differences and applications. We describe Datalift platform, an open source platform to lift raw data sources to semantic interlinked data sources. After comparing Datalift with the Geoknow stack, we apply it in the process of publishing French Administrative Units and French Gazetteer datasets. We then presents the status of the *French LOD*(*FrLOD*) cloud and some sample of queries over structured geometries published within `data.ign.fr` endpoint.

- In the second part of the thesis, we cover three main issues regarding how to present RDF to end-users. First, we make a state of the art review of existing tools and solutions for visual representation and exploration of RDF (Visualbox, LODSpeaKr, Map4RDF, Linked Data Visualization Model,etc.) Then, we present our contribution: the wizard for visualizations including the vocabulary for describing visualizations, the prototype itself, etc. Third, we present two applications applied to events and statistics to showcase the consumption of interlinked datasets in a new fashion. Then, we present a mechanism of extracting and reusing application in open data events. Finally, we provide some insights on revealing the "important" properties of Entities for visualization by reverse engineering the Google Knowledge Panel (GKP). This part is divided in two chapters:

  - **Chapter 4** provides a survey on visualozation tools and applications, with their limitations. We also describe the status of the applications on the web and provide a classification of so-called "Linked Data Applications".
  - In **Chapter 5**, we present our contribution on new approaches to generate visualizations ans applications. We first propose a novel approach for category-based visualizations. We then show an application for geographic domain. Two applications related to events and statistics are also described. Finally, we propose how to improve the discovery of applications contests in Open Data events, through a model and a universal plugin for RDF population.

- In the last part of the thesis in **Chapter 6**, we describe various contributions on the Linked Open Vocabularies (catalog description, vocabulary publications, APIs and endpoints): prefixes harmonization, vocabulary ranking metrics using information content. We also present some insights on checking licenses compatibility between vocabularies and datasets with the defeasible deontic logic by creating an automatic tool for licenses checking.

In **Chapter 7**, we conclude about the presented works, highlight its limitations and suggest new research directions.

Part I

Modeling, Interconnecting and Generating Geodata on the Web

Chapter 2

Geospatial Data on the Web

"The Semantic Geospatial Web will be a significant advancement in the meaningful use of spatial information.[13]
Max J. Egenhofer

2.1 Introduction

The increasing number of initiatives for sharing geographic information on the Web of data has significantly contribute to the interconnection of many data sets exposed as RDF based on the Linked Data principles. Many domains are represented in the Web of data (media, events, academic publications, libraries, cultural heritage, life science, government data, etc.) while DBpedia is the most used dataset for interconnection. For many datasets published, geospatial information is required for rendering data on a map. In the current state of the art, different approaches and vocabularies are used to represent the "features" and their geometric shape although the POINT is the most common representation making use of the latitude/longitude properties defined in the W3C Geo vocabulary. Other geometries from the OpenGIS standard (POLYGON, LINESTRING, etc.) are more rarely exploited (e.g. LinkedGeoData, GeoLinkedData) while fine-grained geometry representations are often required.
In France, the National Geographic Institute (IGN) has started to publish more and more data in RDF, as illustrated by the recent experimental LOD service `http://data.ign.fr`. IGN maintains large databases composed of different types of geographic entities, buildings, topographic information, occupied zones, etc. By reusing existing taxonomies and publishing them on the Web would ease the integration, retrieval and maintenance of French geospatial objects. Moreover, adding semantics to the current data on the Web will not only resolve ambiguities between datasets, but also will enable answering more complex queries than current GIS systems can handle, such as: "*show all buildings used as tribunal courts in the 7th Arrondissement of Paris*". Another use-case is the possibility to reason over parts of a structure: "*show the points where the river Seine touches a boundary of a district in Paris that contain an activity zone*".
In this chapter, we first describe the notion of geographic information, with its specificity and diversity of formats (Section 2.2). Then, we survey the models used on the Web to model existing geodata, by pointing some limitations (Section 2.3 and Section 2.4. We then move to distinguish two levels of georeferencing data (direct and indirect), and the importance of CRSs in the interpretation of geodata (Section

2.5). The contributions start with a REST service for converting geodata in Section 2.6, followed by vocabularies developed both for handling geometries and features on the Web (Section 2.8.1 and Section 2.9). The chapter closes with a take-away message.

2.2 Geographic Information

Geographical phenomena require two descriptors to represent the real world; *what is represent,* and *where it is.* as reported by authors in [17]. For the former, concepts such as "town", "school", "river", are used to recognize the phenomena and described in terms of well-established "objects" or 'entities". At the same time, the type of concepts used to describe a phenomena vary from one scale of resolution to another, depending on the perception of the human observation of the world. The space reference of the phenomena may be defined in terms of geometrically exact or a relative location. The former uses local or world coordinate systems -local or internationally accepted projections that uses geometrical coordinates of latitude and longitude - defined using a standard system of spheroids, projections, and coordinates [17]. Two approaches are generally used to represent geographical primitives in GIS: vector and raster approaches. In the vector data model, the space is represented by a geometry that describes the location and implicitly the shape, and information attributes such as the name, nature, length, or surface. In general, the geometry of a geographic object/entity can be described using three primitives:

- **Point** represented in terms of XY coordinates. For example.
- **Line** represented by a sets of XY coordinate pairs that define a connected path through space.
- **polygon** represented in terms of the XY coordinates of its boundary, or in terms of the set of XY coordinates that are enclosed by such as boundary.

Figure 2.2 shows for different representations of an address in the BD ADRESS® database, such as points, buildings, path, additional address location, etc.

2.2.1 Specificity

Depending on the level of the spatial resolution, a phenomenon reveals more or less details, according to the intended use. So, increasing the level of resolution might reveals internal structure. For example, in the case of a town: sub-districts, suburbs, streets, houses, lamp-posts, traffic signs; which can be important for purpose and not for others. The level of details also influence the representation of a given entity, and thus provides different views of the same entity in a GIS. A town could be represented by a point at a continental level of resolution but as a polygon entity at a regional level. A road at national level is adequately represented by a line; at a street level it becomes an area of paving.

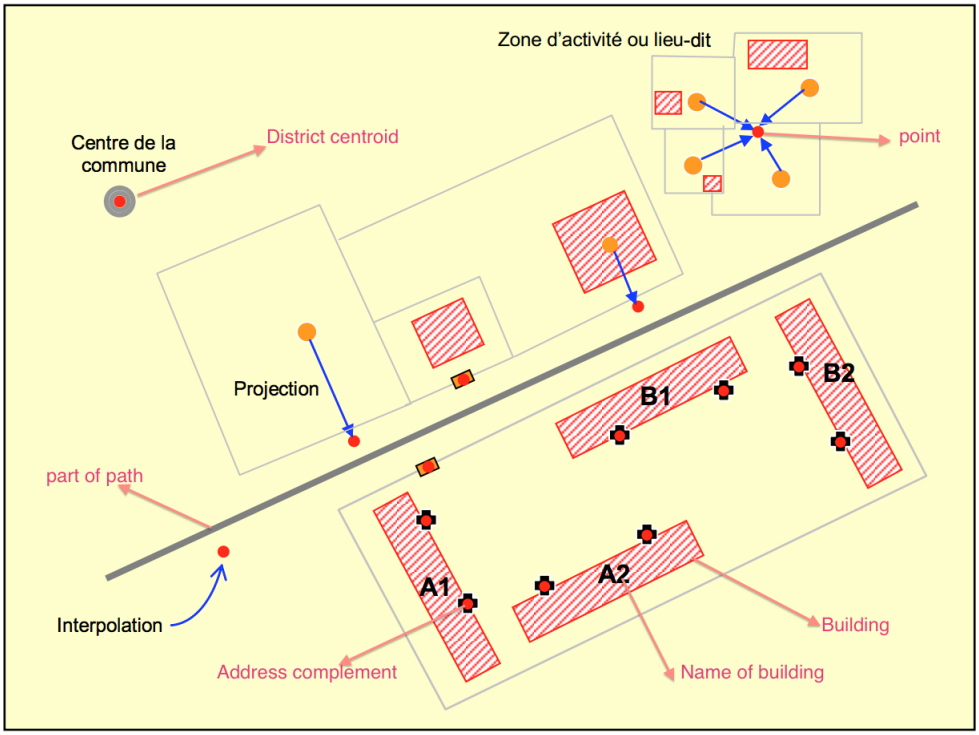


Figure 2.1: Vector representations of entities in BD ADRESSE® produced by IGN-France.

Moreover, geographical data are stored in different GIS according to two different resolution: the metric resolution and "decametric" resolution [18]. Thus, making differences in volume of the data, the scale used to gather and view them, and most importantly their importance. Geodata providers explicitly provides details on the scales and the purpose of their datasets. Let's take the case of BDCARTO® , BDTOPO® and BD ADRESSE® of IGN-France (French mapping agency). BD CARTO® represents the elements using a decametric precision, and contains many themes: roads, administrative units, etc. It helps to manage data from 1 : 50 000 to 1 : 200 000 resolution. BD TOPO is used for producing maps at 1: 25 000 scale, and represents the modelization in 3D of the territory and the amenities with addresses. BD ADRESSE is used for a precised location using the postal code in 1: 25 000 scale.

The aforementioned constraints on abstraction, representation at different scales and requirements of geographical data create some issues both for users and producers. Those issues motivate the need for different relations between GIS datasets. The Web is a good medium to represent a real world phenomena with a unique Uniform Resource Identifier (URI) and associated semantic for referencing, interlinking and tracking the evolution over time.

2.2.2 Data Formats and Serialization

Diverse formats are used to store and exchange geodata in traditional GIS. Some of them are proprietary or closed formats, other are standards defined by OGC. We list below some of the most used formats:

ESRI Shapefile: A shapefile stores non topological geometry and attribute information for the spatial features in a data set. The geometry for a feature is stored as a shape comprising a set of vector coordinates [19]. Regarding this format, four files are processed: the three mandatory dBASE file (.dbf), index file (.shx) and main file (.shp), plus the metadata file (.prj) that describes the CRS used by the dataset. The schema and the type of the thematic and geometric attribute are extracted from the main and dBase file.

Geospatial DBMS: It is a Database that is optimized to store and query geospatial data, such as Oracle Spatial, PostGIS (a spatial extension to PostgreSQL), SpatialLite (a spatial extension to SQLite).

GML: OGC standard encoding specification for geodata in XML that enables the storage, transport, processing and transformation of geographic information. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. As with most XML based grammars, there are two parts to the grammar: the schema that describes the document and the instance document that contains the actual data. A GML document is described using a GML Schema. GML is also an ISO standard (ISO 19136:2007)

Well-Known-Text (WKT): Well-known text (WKT) is a text markup language for representing vector geometry objects on a map, spatial reference systems of spatial objects and transformations between spatial reference systems[1]. The formats were originally defined by the Open Geospatial Consortium (OGC) and described in the specifications for geographic information -simple feature access- [20].

GeoJSON: GeoJSON is a format for encoding a variety of geographic data structures [21]. A GeoJSON object may represent a geometry, a feature, or a collection of features. It also supports the following geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection. Features in GeoJSON contain a geometry object and additional properties, and a feature collection represents a list of features. GeoJSON is the data format of choice for developers which is widely implemented and supported by many tool chains. The default (and strongly recommended) Coordinate Reference System is WGS84, but alternative systems can be specified. The recommended nomenclature

[1] `http://en.wikipedia.org/wiki/Well-known_text`

for CRS systems is to use OGC (Open Geospatial Consortium) URNs, for example urn:ogc:def:crs:OGC::CRS84 (for WGS84). EPSG identifiers, originally from the European Petroleum Survey Group and now maintained by the International Association of Oil and Gas Producers (OGP) can also be used. Alternatively, the parameters for a CRS can be linked to by URL. The example below represents in GeoJSON the location (point) of "Eurecom":

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [43.614151, 7.071414]
  },
  "properties": {
    "name": "EURECOM"
  }
}
```

KML: KML[2] is a language for the visualization of geographic information in 2D (on a map) or 3D (on a globe), including annotation of maps and images. Hence it can be used to specify layers for use in creating maps in a GIS system. A Placemark is one of the most commonly used features in Google Earth. It marks a position on the Earth's surface, using a yellow pushpin as the icon. The simplest Placemark includes only a `<Point>` element, which specifies the location of the Placemark. A Placemark object contains the following elements:

- *A name* that is used as the label for the Placemark
- *A description* that appears in the "balloon" attached to the Placemark
- *A Point* that specifies the position of the Placemark on the Earth's surface (longitude, latitude, and optional altitude)

KML can be used to carry GML content, and GML can be "styled" to KML for the purposes of presentation. KML instances may be transformed loosely to GML, however roughly 90% of GML's structures (such as, metadata, coordinate reference systems, horizontal and vertical datums, etc.) cannot be transformed to KML[3]. KML has very limited support for metadata as recommended by ISO 19115. The CRS in use is implicit and unique.

[2] `http://www.opengeospatial.org/standards/kml`
[3] `http://en.wikipedia.org/wiki/Geography_Markup_Language`

2.3 Status of Vocabularies Usage for Geospatial Data

The publication [22] of statistics concerning the actual usage of vocabularies on the LOD cloud [4] provides not only an overview of best practice usage recommended by Tim Berners-Lee[5], but also provides a rapid view of the vocabularies re-used in various datasets and domains. Concerning the geographic domain, the results show that W3C Geo[6] is the most widely used vocabulary, followed by the `spatialrelations`[7] ontology of Ordnance Survey (OS). At the same time, the analysis reveals that the property `geo:geometry` is used in 1,322,302,221 triples, exceeded only by the properties `rdf:type` (6,251,467,091 triples) and `rdfs:label` (1,586,115,316 triples). This shows the importance of geodata on the web. Table 2.1 summarizes the results for four vocabularies (WGS84, OS spatial relation, Geonames ontology and OS admin geography) where the number of datasets using these vocabularies and the actual number of triples are computed.

Ontologies	#Datasets	#Triples	SPARQL endpoint
W3C Geo	21	15 543 105	LOD cache
OS spatialrelations	10	9 412 167	OS dataset
Geonames ontology	5	8 272 905	LOD cache
UK administrative-geography	3	229 689	OS dataset

Table 2.1: Statistics on the usage of the four main geographic vocabularies (LOD cache should be understood as `http://lod.openlinksw.com/sparql/`). There are many more vocabularies used in the LOD cloud that contain also geographical information but that are never re-used.

2.4 Current Modeling Approach

In this section, we review the different approaches used to model geographical data on the Web, with their advantages and limitations.

2.4.1 Vocabularies for Features

Modeling of features can be grouped into four categories depending on the structure of the data, the intended purpose of the data modeling, and the (re)-use of other resources.

- (i): One way for structuring the features is to define high level codes (generally using a small finite set of codes) corresponding to specific types. Further, sub-types are attached to those codes in the classification. This approach is

[4] `http://stats.lod2.eu`
[5] `http://www.w3.org/DesignIssues/LinkedData.html`
[6] `http://www.w3.org/2003/01/geo/wgs84_pos`
[7] `http://data.ordnancesurvey.co.uk/ontology/spatialrelations`

used in the Geonames ontology[8] for codes and classes (A, H, L, P, R, S, T, U, V), with each of the letter corresponding to a precise category (e.g: A for administrative borders). Classes are then defined as `gn:featureClass a skos:ConceptScheme`, while codes are `gn:featureCode a skos:Concept`.

- (ii): A second approach consists in defining a complete standalone ontology that does not reuse other vocabularies. A top level class is used under which a taxonomy is formed using the `rdfs:subClassOf` property. The LinkedGeoData ontology[9] follows this approach, where the 1294 classes are built around a nucleus of 16 high-level concepts which are: `Aerialway, Aeroway, Amenity, Barrier, Boundary, Highway, Historic, Landuse, Leisure, ManMade, Natural, Place, Power, Route, Tourism, Waterway`. The same approach is used for the French GeOnto ontology (Section 6.4.1), which defined two high-level classes `ArtificialTopographyEntity` and `NaturalTopographyEntity` with a total of 783 classes.

- (iii): A third approach consists in defining several smaller ontologies, one for each sub-domain. An ontology network is built with a central ontology used to interconnect the different other ontologies. One obvious advantage of this approach is the modularity of the conceptualizing which should ease as much as possible the reuse of modular ontologies. Ordnance Survey (OS) follows this approach providing ontologies for administrative regions[10], for statistics decomposition[11] and for postal codes[12]. The `owl:imports` statements are used in the core ontology. Similarly, GeoLinkedData makes use of three different ontologies covering different domains.

- (iv): A fourth approach consists in providing a *nearly flat list* of features or points of interest. This is the approach followed by popular Web APIs such as Foursquare types of venue[13] or Google Place categories[14]. For this last approach, we have built an associated OWL vocabulary composed of alignments with other vocabularies.

2.4.2 Vocabularies for Geometry Shape

The geometry of a point of interest is also modeled in different ways. We complete here the survey started by Salas and Harth [12]:

- *Point representation*: the classical way to represent a location by providing the latitude and longitude in a given coordinate reference system (the most used on the web is the WGS84 datum represented in RDF by the W3C

[8] `http://geonames.org/ontology/ontology_v3.0.rdf`
[9] `http://linkedgeodata.org/ontology`
[10] `http://www.ordnancesurvey.co.uk/ontology/admingeo.owl`
[11] `http://statistics.data.gov.uk/def/administrative-geography`
[12] `http://www.ordnancesurvey.co.uk/ontology/postcode.owl`
[13] `http://aboutfoursquare.com/foursquare-categories/`
[14] `https://developers.google.com/maps/documentation/places/supported_types`

Geo vocabulary). For example, Geonames defines the class `gn:Feature a skos:ConceptScheme` as a `SpatialThing` in the W3C Geo vocabulary.

- *Rectangle* ("bounding box"): which represents a location with two points or four segments making a geo-referenced rectangle. In this way of modeling, the vocabulary provides more properties for each segment. The FAO Geopolitical ontology[15] uses this approach.
- *List of Points*: the geometry shape is a region represented by a collection of points, each of them being described by a unique RDF node identified by a lat/lon value. The `Node` class is used to connect one point of interest with its geometry representation. The POI are modeled either as `Node` or as `Waynode` (surfaces). This approach is followed by LinkedGeoData [10].
- *Sequence of Points*: the geometry shape is represented by a group of RDF resources called a "curve" (similar to LineString of GML). The POI is connected to its geometry by the property `formedBy` and an attribute `order` to specify the position of each node in the sequence. This approach is the one used in GeoLinkedData [11].
- *Literals*: the vocabulary uses a predicate to include the GML representation of the geometry object, which is embedded in RDF as a literal. This approach is followed by Ordance Survey [23].
- *Structured representation*: the geometry shape is represented as a typed resource. In particular, polygons and lines are represented with an RDF collection of basic W3C Geo points. This approach is used by the NeoGeo vocabulary[16].

2.4.3 GeoSPARQL Standard and specifications

OGC-Simple Features Access standard aims to support both representing and querying geospatial data on the Semantic Web. The standard document [24] contains 30 requirements. It also defines a vocabulary for representing geospatial data in RDF and provides an extension to the SPARQL query language for processing geospatial data. Moreover, GEOSPARQL defines functions that request or check properties of a geometry (e.g., isSimple, isEmpty, Dimension, GeometryType, SRID), function that test test topological relations (e.g., contains), and functions that construct new geometries from existing ones (e.g., buffer). The proposed standard follows a modular design with five components:

- (i) A *core component* defining top-level RDFS/OWL classes for spatial objects;
- (ii) a *geometry component* defining RDFS data types for serializing geometry data, RDFS/OWL classes for geometry object types, geometry-related RDF properties, and non-topological spatial query functions for geometry objects;

[15] `http://www.fao.org/countryprofiles/geoinfo/geopolitical/resource/`
[16] `http://geovocab.org/doc/neogeo/`

- (iii) a *geometry topology component* defining topological query functions;
- (iv) a *topological vocabulary component* defining RDF properties for asserting topological relations between spatial objects; and
- (v) a *query rewrite component* defining rules for transforming a simple triple pattern that tests a topological relation between two features into an equivalent query involving concrete geometries and topological query functions.

Geo-Aspect	Requirement	Implementation Definition
Feature	Req 2	The Class `SpatialObject` should be defined & accepted
	Req 3	Defines `Feature rdfs:subClassOf SpatialObject`
	Req 4	Defines 8 Simple Features Object Properties(OP)
	Req 5	Defines 8 Egenhofer OP with domain and range
	Req 6	Defines 8 RCC OP with domain and range
Geometry	Req 7	Defines `Geometry rdfs:subClassOf SpatialObject`
	Req 8	Defines OP `hasGeometry` and `defaultGeometry`
	Req 9	Defines 6 Data Properties: e.g: `dimension, isEmpty, etc.`
Serialization	Req 10-13	`wktLiteral` definitions & URI encoding
	Req 14	Defines `asWKT` to retrieve WKTLiteral
	Req 15-17	GMLLiteral should be accepted
	Req 18	Defines `asGML` to retrieve GMLLiteral

Table 2.2: Requirements and implementations for vocabulary definitions in GeoSPARQL.

Each of the components described above has associated requirements. Concerning the vocabulary requirements, Table 2.2 summarizes the seventeen requirements presented in the GeoSPARQL draft document. GeoSPARQL requires the implementation of functions extensions for Geo-spatial (33) , and provides also corresponding relationships between topological functions. The functions are listed below:

- 8 for RCC: equals(EQ), disconnected(DC), externally connected(EC), partially overlapping(PO), tangential proper part inverse(TPPi), tangential proper part (TPP), non-tangential proper part inverse(NTPPi), non-tangential proper part(NTPP)
- 8 for Simple Features : equals, disjoint, intersects, touches, within, contains, overlaps and cross.
- 8 for Egenhofer relations: egenhofer relations: equals, disjoint, meet, overlpa, covers, coveredBy, inside and contains
- 9 for functions that generate new geometries (Non Topological functions)

Geo-vocabulary	Topological Functions	GeoSPARQL Requirements	Standard Followed
Ordnance Survey Spatial	`easting`, `northing`, `touches`, `within`, `contains`	Part of Req 4	OpenGIS Simple Feature
Ordnance Survey Topography	`contains`, `isContainedIn`	Very small part of Req 4	OpenGIS Simple Feature
Place Ontology	`in`, `overlaps`, `bounded_by`	Small part of Req 4	N/A
NeoGeo Spatial	All RCC8 relations	Part of Req 3; Req 6	Region Connection Calculus (RCC)
NeoGeo Geometry	—	Req 10 - 14	N/A
FAO Geopolitical	`isInGroup`, `hasBorderWith`	–	–
OntoMedia Space	`adjacent-below`, `adjacent-above`, `orbit-around`, `is_boundary-of`, `has-boundary`	–	–

Table 2.3: Comparison of some geo-vocabularies with respect to the GeoSPARQL requirements.

2.4.4 Geospatial Vocabularies and Topological Functions

Based on the GeoSPARQL requirements, we were interested in comparing some geospatial vocabularies[17] to see how far they take already into account topological functions and which are the standard they followed among OpenGIS Simple Features (SF), Region Connection Calculus (RCC) and Egenhofer relations. We find that the NeoGeo (Spatial and Geometry) and OS Spatial vocabularies have integrated in their modeling partial or full aspects of topological functions as summarized in Table 2.3.

As geodata has to be stored in triple stores with efficient geospatial indexing and querying capabilities, we also survey the current state of the art in supporting simple or complex geometries and topological functions compatible with SPARQL 1.1. Table 3.3 shows which triple stores can support part of the GeoSPARQL standard regarding serialization and spatial functions.

[17] `http://labs.mondeca.com/dataset/lov/vocabularySpace_Space.html`

2.5 Georeferencing data on the Web

Georeferencing data either by direct or indirect spatial reference requires some reference datasets that can be used as the spatial frame for anchoring these thematic data. Especially, it requires data on both CRSs and named places, which must be published on the Web of data.

2.5.1 Identifying and describing CRSs on the Web

In order to fulfill the need for CRS identification and description on the Web, OGC maintains a set of URIs for identifying the most commonly used CRS. While very useful, the main disadvantage of this proposal is that the URIs defined by OGC are not very intuitive for users who are not familiar with Spatial Reference System Identifiers defined by geographic information authorities like OGC or EPSG, such as "4326" (which actually refers to a WGS84 CRS defined by the EPSG). Moreover, many CRS commonly used locally, such as deprecated French projected CRS, are not available in that registry. In addition to OGC proposal, several registries have been proposed by the geographic information community for cataloguing existing CRSs. The EPSG Geodetic Parameter Registry[18] allows querying the Geodetic Parameter Dataset gathered by the EPSG. CRSs can be retrieved by name, by code, by type or by coverage area, and their characteristics are displayed on a HTML form. Unfortunately, there is no direct access to these data through dereferenceable URIs.

2.5.2 Direct georeferencing of data on the Web

Modeling direct location information such as coordinates or vector data geometries in RDF still poses some challenges. In [25], we have conducted a survey of the vocabularies used for representing geographical features from vocabularies of feature types to vocabularies for geometric primitives which provide ways for representing extents, shapes and boundaries of those features. Most of vocabularies dedicated to geometry representation reuse W3C Geo vocabulary which allows only WGS84 coordinates, such as NeoGeo[19]. With the rise of the Open Data movement, more and more publishers including governments and local authorities are releasing legacy data that are georeferenced using others CRSs. For example, IGN France releases data using different projected CRSs depending on the geographic extent of each dataset. In order to overcome this limitation on CRSs, the vocabulary designed by OGC GeoSPARQL standard does not reuse W3C Geo vocabulary but proposes another class "Point" instead. Geometries of geographical data represented in RDF with the GeoSPARQL vocabulary are represented by literals encoded consistently with other OGC standards. `gsp:wktLiteral` and `gsp:gmlLiteral` are thus respectively derived from Well-Known Text and GML encoding rules. In `wktLiteral` and `gmlLiteral`, the CRS used to define the coordinates of the point is identified by a

[18] `http://www.epsg-registry.org/`
[19] `http://geovocab.org/doc/neogeo/`

dereferenceable URI which is explicitly stated at the beginning of the literal. This way of associating coordinate reference systems with geometries has the advantage of being consistent with Linked Data principles: each CRS is identified with a dereferencable URI. The main drawback is that such literals cannot be easily queried with SPARQL, unless using regular expression-based filters. To overcome this limitation, we propose in the geometry vocabulary presented in Section 2.8.0.1 to associate each geometry to the CRS used by its coordinates with the property `geom:crs`.

2.5.3 Indirect georeferencing of data on the Web

2.5.3.1 Location Vocabulary

The Location Core Vocabulary [20] provides structure to describe a location in three different ways: by using a place name, a geometry or an address. The vocabulary is heavily based on the definition of IS0 19112 of a location, as "an identifiable geographic place." A part from using simple string labels or names, the vocabulary provides a property to allow a Location to be defined by a URI, such as GeoNames or DBpedia URI. The geographic name used for a spatial object is consistent with the INSPIRE Data Specification on Geographical Names [26]. The Geometry Class denotes the notion of geometry at a conceptual level, and can be encoded in different formats including WKT, GML, KML, RDF+WKT/GML (GeoSPARQL), RDF (WGS84 lat/long, schema.org) and GeoHash URI references. In addition, the geometry property can be associated to either a literal (such as WKT, GML or KML) or a geometry class (e.g., ogc:Geometry and its subclasses, geo:Point, schema:GeoCoordinates and schema:GeoShape, a GeoHash URI reference). However, the CRS identifier of the geometry is either embedded in the literal (e.g., WKT, GML) or implicit in the more structured serialization (e.g., WGS84 lat/-long), schema.org, GeoHash).

2.5.3.2 Datasets using indirect georeferencing

Modeling indirect location information such as administrative units or named points of interest in RDF is preferably done by identifying such geographic features with URIs and describing them by their properties, so that they can be referenced by other datasets. This is the case in one of the most reused datasets of the Web of data, namely Geonames[21]. However, there are yet very few reference datasets for the French territory on the Web of data. A simple example is the current resource for *Paris* in the French DBpedia[22]. The department's name associated to this resource is a literal named "Paris" and the different arrondissements composing the city are modeled as `skos:Concept` instead of `dbpedia-owl:Place`. Even Geonames data remain very limited, as French administrative units are provided

[20] `http://www.w3.org/ns/locn`
[21] `http://sws.geonames.org/`
[22] `http://fr.dbpedia.org/resource/Paris`

as simple geometries (POINT). The "Official Geographic Code"[23] published by the French Statistical Institute (INSEE) is the most up-to-date and accurate dataset on French administrative units, but unfortunately it contains no geometrical description of their boundaries. This has the consequence of not having a baseline during mapping process for application developers trying to consume specific data coming from France. Datasets describing administrative units, points of interest or postal addresses with their labels and geometries, and identifying these features with URIs could be used with benefits not only for georeferencing other datasets, but also for interlinking datasets georeferenced by direct and indirect location information.

2.6 A REST Service for Converting Geo Data

2.6.1 Datum

The Earth is shaped like a flattened sphere. This shape is called an ellipsoid. A datum is a model of the earth that is used in mapping. The datum consists of a series of numbers that define the shape and size of the ellipsoid and it's orientation in space. A datum is chosen to give the best possible fit to the true shape of the Earth. There are a large number of datum in use. Many of them are optimized for use in one particular part of the world. An example is the Geodetic 1949 datum that has been used in New Zealand. Another example, familiar to GPS users, is the WGS-84 datum. WGS-84 is an example of a datum that is used globally. A point (location) is referenced by its longitude and latitude values. Longitude and latitude are angles measured from the earth's center to a point on the earth's surface. The angles often are measured in degrees (or in grads). It is important to keep in mind that latitude and longitude are always specified in terms of a datum. The latitude and longitude of one current position are different for different datums. For example, the *"Théâtre National de Nice"* in Nice, France is at *43.700594° N, 7.277959° E* in WGS 84 coordinates and *43.700570° N, 4.941204° E* in NTF coordinates, an old France CRS. If the latter coordinates are used in WGS 84, they will point to a position which is approximately *295.17 kilometers* away from the theater. So when working with latitude/longitude coordinates and getting an error of a couple of hundred meters, it is most likely that the wrong datum is in used.

2.6.2 Tools for converting Datum

As we have seen, geodata interpretation relies on a coordinate reference system, and while the WGS84 CRS is the *de-facto* standard for GPS devices, many other CRS are in used. For example LAMBERT 93, RGM 04 or RGR 92 are respectively used for georeferencing points of interests in France continental, Mayotte or La Reunion. We have developed a REST service that is capable of transforming one dataset using a particular CRS into another one.

[23] `http://rdf.insee.fr/sparql`

A software called Circé[24], published by IGN, provides the abilities to convert coordinate between CRSs in France and WGS 84. It has two conversions mode: standard and grid. In both modes, the user is required to input the source CRS values in order to convert. Based on what datum and projection method used, the number of required fields are different. Circé also has a batch converting mode which is done by supplying a file. The format of the content of the file is simple. The available formats are converted into *[location name] Lat/Lon Lon/Lat [Altitude/Height]*. There is an option to choose which format to use. But due to the fact that it is a closed source software, no one can use the software as a service for their system.
Another tool is a Web based application called the world coordinate converter at `http://twcc.free.fr`. This tool can convert between numerous of CRSs. Unlike Circé which only supports France and WGS 84, this tool supports conversion of international CRSs and nationals' CRSs. The result output from these two tools are the same. However, just like Circé, no API or any open service for the community to use their full power horse unless going to their website and use it like an application. These tools are great as a standalone tool for end user, but not so great for developer community. The developer can only use them to test their result. There is no possible way to use their fully function algorithm unless develop it again.

Our proposal: The purpose of the REST Converter is to propose a web based service to perform conversion between various CRSs.
The algorithms implemented are the ones described at `http://geodesie.ign.fr/index.php?page=algorithmes` and available within the standalone Circé software[25].
At the moment, the following features are implemented in the Geo Converter:

- from/to WGS 84 to/from WGS 84 UTM ;
- from/to WGS 84 to/from Lambert 93 and
- from/to WGS 84 UTM to/from Lambert 93

The API can also convert a file with space separated values. The API supports JSON as one of the output format. The code of the REST service is available at `https://github.com/vienlam/Geo`.

2.6.3 Algorithms Evaluation

Figures Fig.2.2 and Fig.2.3 show the resulting conversion from our tool, Geo Converter, in comparison with Circé and twcc.free.fr (TWCC).
The results show that the results from Geo Converter are not too much different from Circé and TWCC. This deviation can be tolerated, since when showing on the map, they are basically the same point. There is no need for evaluation of conversion between Lambert 93 to UTM, since it is needed an intermediate step of convert to

[24] `http://geodesie.ign.fr/?p=53&page=circe`
[25] `http://fr.wikipedia.org/wiki/Circe_(logiciel)`

Conversion from WGS 84 to Lambert 93				
Tools	**Input**		**Output**	
	Latitude (DD)	**Longitude (DD)**	**X (m)**	**Y (m)**
Geo Converter	43.700594	7.277959	1044752.61	6298403.61
Circé			1044752.20	6298404.17
twcc.free.fr			1044752.61	6298403.61
Conversion from Lambert 93 to WGS 84				
Tools	**Input**		**Output**	
	X (m)	**Y (m)**	**Latitude (DD)**	**Longitude (DD)**
Geo Converter	1044752.61	6298403.61	43.700594	7.277959
Circé			43.700589	7.277964
twcc.free.fr			43.700594	7.277959

Figure 2.2: Results of conversion from WGS 84 to Lambert 93. Note: DD=Decimal Degree

Conversion from WGS 84 to WGS 84 UTM						
Tools	**Input**		**Output**			
	Latitude (DD)	**Longitude (DD)**	**X (m)**	**Y (m)**	**Hem**	**Zone**
Geo Converter	43.700594	7.277959	361243.56	4840060.20	N	32
Circé			361243.52	4840060.21	N	32
twcc.free.fr			361243.52	4840060.21	N	32
Conversion from WGS 84 UTM to WGS 84						
Tools	**Input**				**Output**	
	X (m)	**Y (m)**	**Hem**	**Zone**	**Latitude (DD)**	**Longitude (DD)**
Geo Converter	361243.52	4840060.21	N	32	43.700586	7.277959
Circé					43.700594	7.277959
twcc.free.fr					43.700594	7.277959

Figure 2.3: Results of conversion from WGS 84 to WGS 84 UTM. DD=Decimal Degree

WGS 84. The conversion from Lambert 93 to WGS 84 works well, as well as from WGS 84 to WGS UTM.

2.6.4 API Access and Parameters

2.6.4.1 Simple Converter

The API is working through URL like any RESTful service does. The syntax for the service is (note that {} is required and [] is optional): `http://{domainname}/eurecom.geo.rest/api/converter/{D1}[P1]{D2}[P2]?{Parameters}`
Where:
`domainname`: Eurecom domain name.
`D1` and `D2`: Datum of source and target CRS respectively.
`P1` and `P2`: Projection of source and target CRS respectively.
`Parameters:` The required parameters as input to the service. See table below for detail of required parameters for each converter. Parameters are provided as p1=v1&p2=v1 where p1 is the first parameter with v1 is its value, etc.
An example can be:

`eurecom.fr/eurecom.geo.rest/api/converter/WGS84RGF93Lambert93?lon=4.7021484375&lat=45.2130035559939`

2.6.4.2 Batch Converter

The API also support batch converter by using file. The URL syntax is:

`http://{domainname}/eurecom.geo.rest/api/converter/file/{D1}[P1]{D2}[P2]?{Parameters}`

Where D1, P1, D2, and P2 is as before. The parameters are the source file, and the encoding system. The order of the input coordinates in the file is matter, so the exact order is as follow: longitude/x latitude/y [zone] [hemisphere] The first value should be longitude, in case geographic coordinate, or x, in case of planimetric coordinate. The second value is latitude or y. In case of UTM coordinates, the following third value is the zone. The last value is hemisphere.

2.6.4.3 Result Format

Normally, the API will return the result in form of normal string with the space character as delimiter for each coordinate. In case of batch converter, each location will be in one line. For example:

`eurecom.fr/eurecom.geo.rest/api/converter/WGS84RGF93Lambert93?lon=4.7021484375&lat=45.2130035559939`

will return a string of:

`833607.9336802219 6458515.660215093`

The order of value respond to the order of coordinate as follow:

`{longitude/x} {latitude/y} [zone] [hemisphere]`

JSON format result: However, one can ask the API to return the result string in JSON format. To demand the API to do so, simply put the json parameter with the value 1 to the link. For example:

`eurecom.fr/eurecom.geo.rest/api/converter/WGS84RGF93Lambert93?lon=4.7021484375&lat=45.2130035559939&json=1`

will return a string in JSON format (no line breaking):

```
{"y":6458515.660215093,"x":833607.9336802219}
```

JSON format return can also be applied for batch converter. The command is as before. An example of JSON format return of batch converter from WGS84 to Lambert93:

```
1 {"point1":{"y":6543019.59988031,"x":882408.2999938729},"point2":{"y"
      :6544401.599880268,"x":881947.5999938913},"point3":{"y"
      :6581538.799879094,"x":849722.3999950405},"point4":{"y"
      :6561282.999879616,"x":917481.0999927416},"point5":{"y"
      :6561139.999879618,"x":917474.5999927415}}
```

Listing 2.1: Sample output of the batch converter

For better human readable result, a tool such as JSONLint[26] can be used for the JSON format.

2.6.5 User Interface

To access the User Interface (UI), the URL is: `http://{domainname}/eurecom.geo.rest/`. Figure 2.4 shows the landing page of the converter. The UI shows two systems, which are the source and the target system, to provide input. Which system is source or target depend on which button at the end is used. If the button *Convert S1-S2* is clicked, then the System 1 will be the source and System 2 is the target, and *vice versa* for button *Convert S2-S1*. The required inputs is just as mentioned before. First is the datum to use. Next is the projection method. If no project method is used then choose None. Next is the inputs of the source system. The map under the form will show where the coordinate in the source system point to through a marker. This marker can be used as an input to the source system as well. By dragging the marker, the system will update the input values of the source system, which is the system that have the last edited field or act as source system in last conversion, to correspond to the marker position. Furthermore, the "Use Marker's Position" button can be used to take marker's position and convert it to the system which the button resides in. This will not change the source system.

2.7 Best Practices for Modeling Geospatial vocabularies

In [25], we already surveyed numerous vocabularies for representing geographical features and their geometries, either using a literal (e.g. wktLiteral) or a structured

[26] `http://jsonlint.com/`

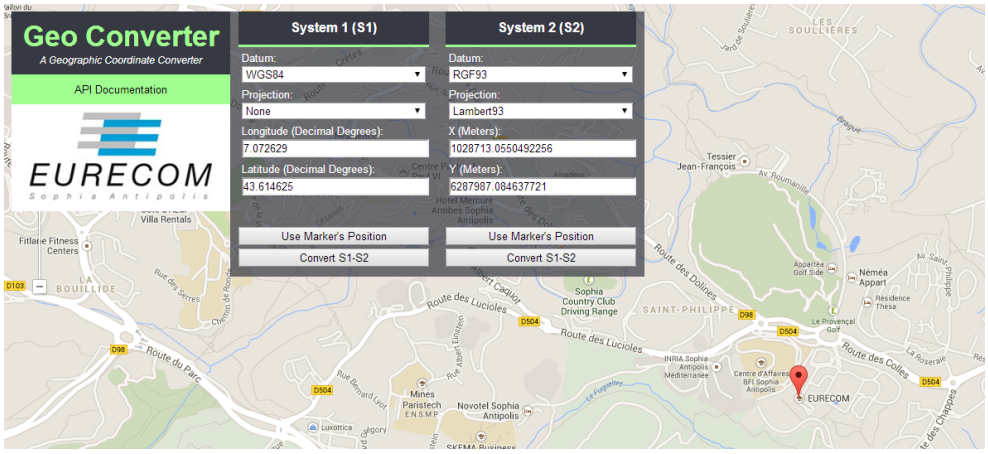


Figure 2.4: The User Interface of the Geo Converter

representation à la NeoGeo. We concluded the survey with some recommendations for geometry descriptions:

- the distinction of geometry versus feature and a property linking both classes (e.g. for attaching provenance information on how some points of a geometry have been collected),
- the ability to represent structured geometries (e.g. for performing simple spatial queries on the data, even when they are stored in a triple store that do not implement the GeoSPARQL standard),
- the integration of any coordinate reference system (e.g. for allowing projected coordinates for cartographic purposes).

In addition to these recommendations, we also think that the domain of the property used to link a feature to its geometry should be left empty in order to accept links between any type of resource and a geometry. This would be useful for example, to associate a person to the coordinates of their birthplace.

2.7.1 Some Recommendations

The alignment of existing taxonomies for describing geodata enables interoperability of symbolic descriptions. The need for a better choice of geometric structure, typically the choice between literal versus structured representations depends on three criteria:

- (i) the coverage of all the complex geometries as they appear in the data;
- (ii) a rapid mechanism for connecting "features" to their respective "geometry";
- (iii) the possibility to serialize geodata into traditional formats used in GIS applications (GML, KML, etc.) and

- (iv) the choice of triple stores supporting as many as possible functions to perform quantitative reasoning on geodata.

It is clear that a trade-off should be made depending on the technological infrastructure (e.g: data storage capacity, further reasoning on specific points on a complex geometry). The following points helps understanding better some of the challenges:

- **Complex Geometry Coverage:** We have seen that on the Web of Data, there are few modeling of geodata with their correct shape represented as a LINE or POLYGON. However, some content providers (e.g. IGN) need to publish all types of geodata including complex geometries representing roads, rivers, administrative regions, etc. Two representations are suitable: *OS Spatial* and *NeoGeo* ontologies (Table 2.2). Direct representation of the GeoSPARQL vocabulary is also suitable.

- **Features connected to Geometry:** In modeling geodata, we advocate a clear separation between the features and their geometry. This is consistent with the consensus obtained from the different GeoVocamps[27] and the outcome of this approach is expressed in the modeling design of NeoGeo. The top level classes `spatial:Feature` and `geom:Geometry`are connected with the property `geom:geometry`.

- **Literal versus structured Geometry:** Decomposing a LINE or a POLYGON into multiple results in an "explosion" in the size of the dataset and the creation of numerous blank nodes. However, sharing points between descriptions is a use case with such a need. IGN has such use-cases and the natural solution at this stage is to consider reusing the NeoGeo ontology . The choice of the triple store (e.g.,Virtuoso vs Open Sahara) is not really an issue, as the IndexingSail[28] service could also be wrapped on-top of Virtuoso to support full OpenGIS Simple Features functions[29].

2.8 Vocabularies for Geometries and Feature Types

Direct georeferencing of data implies representing coordinates or geometries and associating them to a CRS. This requires vocabularies for geometries and CRSs. Besides, indirect georeferencing of data implies associating them to other data on named places. Preferably, these data on named places should be also georeferenced by coordinates in order to serve as basis for data linking between indirectly and directly georeferenced datasets. In this section, we present the vocabularies that we have defined and reused for geographic data publishing. This requires reference geographic data on named places and therefore vocabularies for describing feature types and their properties.

[27] `http://www.vocamp.org`
[28] `https://dev.opensahara.com/projects/useekm/wiki/IndexingSail`
[29] `http://www.opengeospatial.org/standards/sfs`

2.8.0.1 A vocabulary for geometries

On the current usage of georeferencing resources on the Web of data, it is assumed that the coordinates should be in WGS84, and hence the definition of the point. However, publishers might have data in different CRSs according to the location. Thus, our proposal is to define a more generic class for a POINT with the benefit of choosing the CRS of the underlying data, as depicted in the Listing 2.3. The naming convention used for the `geom` vocabulary follows the terms used by the Simple Features vocabulary. The French translation of terms is based on the glossary of multilingual terminology of ISO/TC 211 available at `http://www.isotc211.org/Terminology.htm`.

Axiom 1 *: A resource of type `geom:Geometry` should be associated to exactly one resource of type `ignf:CRS` via the property `geom:crs`.*

Axiom 2 *: **A POINT** is a subclass of a **GEOMETRY**.*

Regarding alignments with some existing vocabularies, the class `geom:Geometry` is a subclass of both `sf:Geometry` and `ngeo:Geometry`. The class contains in addition the property `geom:crs`. Moreover, it is possible to obtain equivalences between data modeled with *ngeo* vocabulary and *geom* vocabulary. The following SPARQL query make it possible:

```
1
2 CONSTRUCT {
3     [] a geom:Point;
4         owl:sameAs   ngeo:Point.
5
6 } WHERE {
7   [] a geom:Geometry;
8    geom:Point;
9    geom:systCoord
10 <http://data.ign.fr/id/ignf/crs/WGS84G>.
11 }
```

Listing 2.2: SPAQRL Query for creating sameAs links between data modeled with *ngeo* and *geom* vocabularies

Axiom 3 *: An instance of the class `geom:Point` is associated with exactly one instance of `ignf:CRS` via the property `geom:crs`. An instance of a `geom:Point` has exactly one coordinate X and exactly one coordinate Y. The coordinates are `xsd:double` and referred to the following properties:*

- *`geom:coordX` refers to, in an ellipsoidal CRS, the longitude of a point and within a projected CRS, the value of false easting of a point.*
- *`geom:coordY` refers to, in an ellipsoidal CRS, the latitude of a point and within a projected CRS, the value of false northing of a point.*

```
geom:Point a owl:Class;
  rdfs:label "Point"@en, "Point"@fr;
  rdfs:subClassOf geom:Geometry;
  owl:equivalentClass
    [a owl:Class ;
    owl:intersectionOf
    ([a owl:Restriction;
       owl:onDataRange xsd:double;
       owl:onProperty geom:coordY;
         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger]
         [a owl:Restriction;
       owl:onDataRange xsd:double;
       owl:onProperty geom:coordX;
       owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger])
     ] ;
     rdfs:subClassOf sf:Point.
```

Listing 2.3: Definition in Turtle of the axiom defining a POINT.

Axiom 4 *(PointsList): A* `geom:PointsList` *is a subclass of rdf:List. An instance of* `geom:PointsList` *is composed of only instances of type* `geom:Point`*.*

2.8.0.2 Extending GeoSPARQL vocabulary

In order to fulfill these recommendations, we have developed a new vocabulary that re-uses and extends the existing vocabularies for representing geometries, namely:

- `http://www.opengis.net/ont/geosparql#` (prefix `gsp`). This vocabulary provides the basic concepts to represent geographical data such as SpatialObject, Feature or `Geometry`. A Feature is linked to a Geometry via the relation `gsp:hasGeometry`. The geometries are typed strings (`gsp:gmlLiteral` or `gsp:wktLiteral` corresponding respectively to the properties `gsp:asGML` and `gsp:asWKT`). The vocabulary contains also spatial functions.
- `http://www.opengis.net/ont/sf#` (prefix `sf`): This vocabulary is based on the OGC standard Simple Features Access [27]. The class `sf:Geometry` is a subclass of `gsp:Geometry`.

Reusing and extending GeoSPARQL Simple Features vocabulary with structured geometries à la NeoGeo enables us to represent geometries both with GeoSPARQL compliant literals and with structured geometries that can be handled easily with SPARQL. The extension for structured geometries consists in defining a subclass for each class from the `sf` vocabulary, and defining properties to associate its instances with a CRS and coordinates or other suitable geometric primitives. For example, the class `geom:Point` is a subclass of `sf:Point`. An instance of `geom:Point` is associated with exactly one instance of `ignf:CRS` via the property `geom:crs`, and it has exactly one coordinate X and exactly one coordinate Y. It can also have a Z coordinate. The coordinates are `xsd:double` and correspond to the properties `geom:coordX:`, `geom:coordY:` and `geom:coordZ:` respectively. Other complex

geometries are also defined, such as Linestrings, LinearRings, Polygons or MultiPolygons. Their definitions are based on the class `geom:Point`. As an example, an instance of `geom:Linestring` is defined as an instance of `geom:PointsList` which is an ordered `rdf:List` of instances of `geom:Point` designated by the property `geom:points`.
We have also defined a property `geom:geometry` with an empty domain. Thus, our proposal defines a more generic class for a POINT with the benefit of choosing the CRS of the underlying data. Figure 2.5 gives an overview of the relationships between the high level concepts with geometries, CRS and topographic features.

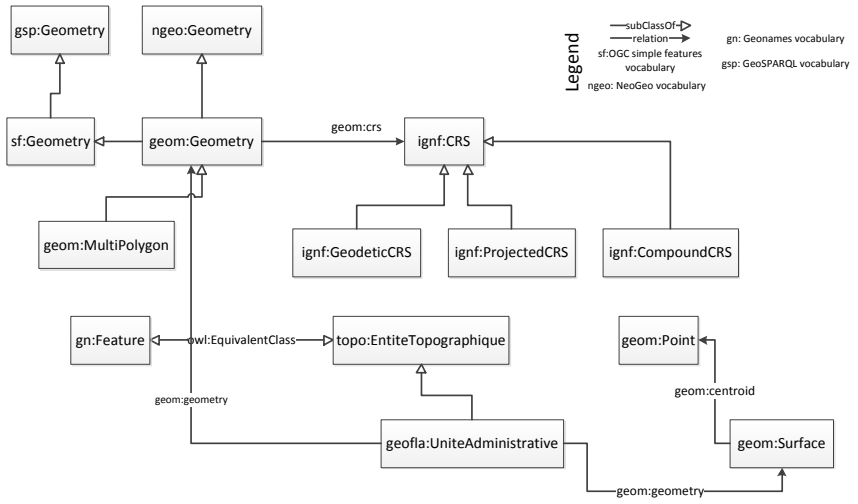


Figure 2.5: High level classes of ignf, geom and topo vocabularies; relationships between them and mappings with external vocabularies.

2.8.1 A vocabulary for Topographic entities

A topographic entity `topo:EntiteTopographique` is the class associated to a phenomenon with an associated location on the Earth[30]. `topo`[31] vocabulary contains 8 direct subclasses: Buildings and structures, Cableway transport line, Energy transport infrastructure, Inland hydrographic feature, Relief feature, Road transport feature, Transport by rail feature, Vegetation area and Working area of interest. Furthermore, we use a second level of classes to model direct subclasses of the previous ones. For example, a Building is specialized in 6 different classes: Cemetery, Sports ground, Structure, Tank and Taxiway. A building is then connected with the property `topo:typeDeBatiment` (type of building) to different *'nature'* of buildings,

[30]The names used in the description of the classes or properties are rdfs:label[@lang='en'] of the actual URIs in the models.
[31]`http://data.ign.fr/def/topo`

which are modeled as SKOS concepts. SKOS is intensively used to easily group concepts into different schemes (using `skos:hasTopConcept`) and provide semantic relationships (e.g: `skos:broader`, `skos:narrowMatch`) among them. This gives flexibility in the model by defining few high level classes and restrictions for the properties. Table 2.4 gives a listing of the different SKOS namespaces defined to capture some high level concepts.

URL	Description
cdtopo:typedezai:liste	list of different areas of activities and interest
cdtopo:typedebatiment:liste	list of types of buildings
cdtopo:typedeterraindesport:liste	list of types of sports ground
cdtopo:typedeconstruction:liste	list of types of construction
cdtopo:typedereservoir:liste	list of types of tanks
cdtopo:typedevegetation:liste	list of types of vegetation
cdtopo:typederelief:liste	list of types of relief
cdtopo:typedevoieferree:liste	list of types of railway track
cdtopo:typedetransportcable:liste	list of types of cableway transport
cdtopo:typedefranchissement:liste	list of types of crossing
cdtopo:typederoute:liste	list of types of road
cdtopo:typedereservoir:liste	list of types of waterhole
cdtopo:typedelaisse:liste	list of types of tide line

Table 2.4: List of concept schemes used in the topographic ontology.

2.8.2 Publishing structured geometries from geographic data

The vocabulary for geometry reused by a geodata converter that takes traditional GIS data as input and outputs RDF data with geometries defined both with a `gsp:wktLiteral` and with a structured representation compliant with our vocabulary. Geometries are automatically associated with the chosen CRS. This converter is implemented as a plugin of the Datalift platform (cf. Section 3.3) and can be reused easily for geographic data publishing purpose.

2.8.3 CRS requirements for the French territory

As explained in Section 1.1, making explicit the CRS used in a given dataset is a very important issue when dealing with direct location data. This is especially important in the field of geographical information where different CRSs are commonly used due to technical or legal requirements. For INSPIRE Directive, CRS are considered as reference data used for linking thematic data [26], and must be described according to ISO 19111 standard. To be consistent with Linked Data principles, CRS should be identified by URIs, like in OGC proposal. Moreover, as Linked Data users are not always familiar with CRS identifiers commonly used within the geographic

information community, URI used to identify CRS should use more intuitive names. Finally, consistently with our goal of contributing to a better georeferencing of data on the French territory, we need an access to the descriptions of all French CRSs, including some deprecated but still used CRSs like "Lambert 1".

Prefix	URI
geofla	http://data.ign.fr/def/geofla#
geom	http://data.ign.fr/def/geometrie#
ignf	http://data.ign.fr/def/ignf#
rgeofla	http://data.ign.fr/id/geofla/
topo	http://data.ign.fr/def/topo#
rtopo	http://data.ign.fr/id/topo/

Table 2.5: URI schemes and conventions used for vocabularies and resources.

The Information and Service System for European Coordinate Reference Systems[32] provides an access to ISO 19111 standard-based descriptions of the main European CRSs but has the same limitation as the EPSG registry: access to the descriptions is not allowed by URI, but only through a cartographic interface. `SpatialReference.org` initiative aims at allowing users to use URI-based references to spatial reference systems, including some CRSs defined and maintained by IGN France. Besides, the proposed URL policy is not very intuitive. As an example, this URL identifies the projected system defined by IGN France, Lambert 93: `http://spatialreference.org/ref/sr-org/7527/`. Moreover, the definitions of some deprecated CRSs such as Lambert zone projected CRSs (which are still used in some datasets) seem to be referenced only for the authority EPSG and not for IGNF, which also maintains a registry of CRSs. ISO 19111 standard-based definitions of all CRSs defined and maintained by IGN France are published in an XML file[33]. References to equivalent definitions provided by the EPSG registry are explicitly stated with EPSG SRID. CRSs are identified by URIs using short names instead of numeric codes. For example, `http://registre.ign.fr/ign/IGNF/crs/NTFLAMB2E` is the URI designed for the "Lambert 2 étendu" projected system. Indeed "NTFLAMB2E" is used to identify the projected system "Lambert 2 étendu" which is based on NTF (New French Triangulation) geodetic reference system. Unfortunately, this registry is still in evolution and its URIs are not dereferenceable yet.

As no existing registry fulfilled all our requirements, we have developed a vocabulary[34], inspired from the ISO 19111 schema for CRSs description. Then we have converted IGNF CRSs registry into RDF, and published this dataset on the Web with the Datalift platform[35]. Therefore, the description of the "NTF Lambert 2 étendu" projected CRS can be retrieved at this URL `http://data.ign.fr/id/`

[32] `http://www.crs-geo.eu`

[33] `http://librairies.ign.fr/geoportail/resources/IGNF.xml`

[34] `http://data.ign.fr/def/ignf`

[35] A service to lookup CRS in RDF can be found at `http://www.eurecom.fr/~atemezin/ignf-lookup/`

REGION	COORDINATE SYSTEM	ELLIPSOID	PROJECTION SYSTEM	ALTIMETRY SYSTEM
FRANCE METROPOLITAN	RGF93	IAG GRS 1980	Lambert 93 and CC 9 Zones	
MAYOTTE	RGM04 (ITRF2000)	IAG GRS 1980	UTM 38 South	SHOM 1953
GUYANE	RGFG95	IAG-GRS 1980	UTM 21 22 North	
MARTINIQUE	WGS84	IAG-GRS 1980	UTM 20 North	
GUADELOUPE	WGS84	IAG-GRS 1980	UTM 20 North	
LA RÉUNION	RGR92	IAG-GRS 1980	UTM 40 South	GGR 99
NOUVELLE-CALÉDONIE	ITRF90	IAG-GRS 1980		
POLYNÉSIE	RGPF	IAG-GRS 1980	UTM 5, 6, 7 and 8 South	Tahiti IGN 1966
WALLIS ET FUTUNA	MOP87	International 1924		
SAINT-PIERRE ET MIQUELON	RGM01 (ITRF2000)	IAG GRS 1980	UTM 21 North	Danger 1950
ILE CLIPPERTON	Marine 1967	International	UTM 12 South	

Figure 2.6: Coordinate Reference Systems used in France

`ignf/crs/NTFLAMB2E`.

2.9 Vocabularies for Geographic Feature Types

Indirect georeferencing of resources on the Web requires reference geographic data on named places and therefore vocabularies for describing feature types and their properties. Therefore, we have chosen to publish a reference dataset on administrative units called GEOFLA®, which is already available in GIS format under an Open Data license. We have also made tests of data conversion and interlinking with another largest dataset on French names places. We have produced and published two vocabularies to describe these datasets, to make sure that all concepts and properties needed would be available. In the GEOFLA® vocabulary, 5 classes have been defined: commune, canton, arrondissement, department and region. In the BD TOPO® vocabulary[36] 35 main classes have been defined. They represent the main types of geographic features represented in the BD TOPO® database. In both vocabularies, properties have been defined based on the attributes of their related classes in the databases. The geographic feature types defined as values of attributes "nature" are modeled as instances of `skos:Concept`. SKOS is intensively used to easily group concepts into different schemes (using `skos:hasTopConcept`) and provide semantic relationships (e.g: `skos:broader`, `skos:narrowMatch`) among them. We also provide alignments with Geonames vocabulary, where `topo:Place`

[36] `http://data.ign.fr/def/topo`

is subclass of `gn:S` and `owl:sameAs` linked concepts.[37]
All the classes are defined as subclasses of `topo:EntiteTopographique` which defines the representation of a real world entity associated to a location relative to the Earth, consistently with ISO TC 211 and OGC standards. The GEOFLA®'s application schema is composed of classes representing different types of french administrative units, namely communes, cantons, arrondissements and departments. In `geofla` vocabulary, we add a class `Region` from the instances of the class department via two attributes that precise to what region each instance of department belongs. Their properties are defined based on the attributes of their related classes in GEOFLA® database.
A Commune has an attribute called *"nature"* whose enumerated values precise whether the commune is the capital of a bigger administrative unit, modeled in the vocabulary by the ObjectProperty `geofla:statut` with range `skos:Concept` defined in this specific `skos:ConceptScheme`
`http://{BASE}/codes/geofla/typedecommune/liste` pointing to the different types of French administrative unit's capital.

2.10 Summary

We have presented in this chapter the different vocabularies used to model geospatial data on the Web, based on direct and indirect georeferencing. After identifying one limitation on the lack of an explicit CRS reference on the data, e then proposed a REST service for converting between different CRSs. Then, we developed three vocabularies (geom, ingf, topo) extending existing ones and integrating two additional advantages: an explicit use of CRS identified by URIs for geometry, and the ability to describe structured geometries in RDF. In the next chapter, we will go through the conversion and the publication of geospatial data on the Web.

[37] `https://github.com/gatemezing/ign-iswc2014/blob/master/vocabularies/mappingsGeonames.ttl`

Chapter 3

Publishing and Querying Geodata

"If you're a geospatial developer, AJAX is not a domestic cleaning product. If you're a web dev, a polygon is not a dead parrot"[1].
Steve Peters
(UK Government's Department
for Communities and Local Government)

3.1 Introduction

Geospatial data need tools for converting native formats into RDF. Those tools are based on OGC libraries for extracting features, and vocabularies provided to build the RDF. Once the dataset obtained, it has to be stored in an efficient way, interconnected to other datasets, and then consumed using SPARQL queries. Moreover, all those steps might be integrated in frameworks that ease the overall process of publishing geodata. In this chapter, we discuss the aforementioned topics, each time surveying the state-of-art and highlighting our contributions.

3.2 Existing Tools for Converting Geospatial Data

To address the need for converting geodata into a graph model such as RDF, there are some tools that have been proposed to generate RDF data from legacy geospatial datasets. The differences among the tools are based on four main criteria :

- **Input format:** The different types of formats accepted as input of the tool;
- **Vocabulary:** The vocabulary used to handle the geometry aspect of the spatial data during the RDF conversion step;
- **CRS converter:** The presence or not of a CRS converter between different CRSs in the final output of the tool ;
- **Output:** The type of serialization in RDF, and more importantly the choice between structured geometry, OGC compliant (WKT, GML literals) or both for the geometry part of the features.

Almost all the tools use the OGC libraries for parsing and extracting features from shape formats, such as GDAL (Geospatial Data Abstraction Library) and GeoTools.

[1] `http://www.w3.org/2014/03/lgd/report`

3.2.1 Geometry2RDF

Geometry2RDF [11] is a java based tool[2] that generates RDF triples from geometrical information, which can be available in GML or WKT. The tool takes as input any ESRI shapefiles, spatial DBMS (Oracle, PostgreSQL, MySQL, etc), transforms into GML (using GeoTools[3]) and then generates RDF (using Jena[4]) consistent with the NeoGeo vocabulary. The default CRS used for the geometry is WGS84. The architecture is flexible to run as a standalone platform or as a library.

3.2.2 TripleGeo

TripleGeo [28] is an ETL (Extract- Transform-Load) tool derived from Geometry2RDF (Cf. Section 3.2.1) to transform a variety of geospatial databases and shapefiles (including KML and INSPIRE compliant files) into RDF triples. Triples can be exported according to the GeoSPAQL standard, the WGS84 vocabulary and the Virtuoso RDF vocabulary[5]. In addition TripleGeo allows on-the-fly reprojection between CRSs, e.g., transform geometries from GreekGrid87 (a local CRS) into WGS84 (used for GPS locations). The code is available at `https://github.com/GeoKnow/TripleGeo`.

3.2.3 shp2GeoSPARQL

shp2GeoSPARQL [29] is also an extension of Geometry2DF which transforms Shapefiles into RDF in the cadastral domain using ISO 19152 [30] (Land Administration Domain Model) and GeoSPARQL. The geometries obtained by shp2GeoSPARQL are consistent with the GeoSPARQL geometry vocabulary.

3.2.4 Limitations of existing tools

Currently, the tools achieving the transformations of geospatial data into RDF still suffer from some limitations. On one hand, they are all compatible with GeoSPARQL standard, with in turns has a handful endpoints that implements all the requirements. Moreover, geometries are literals (e.g., wktLiteral, gmlLiteral) with embedded CRS. On the other hand, tools based on NeoGeo vocabulary outputs structured geometries that can be easily handled by existing Triple Stores and SPARQL queries. However, NeoGeo only allows geometry in WGS84 CRS. The ideal scenario could be a tool that provides output consistent with both GeoSPARQL and structured geometries handling multiple CRSs. Figure 3.2.4 shows the generic architecture for tools to convert ESRI ShapeFiles into different flavor of RDF. Tools differ mainly in the presence or not of the CRS reprojection, the vocabularies used for the RDF output, and the compatibility with GML and WKT representations.

[2] `https://github.com/boricles/geometry2rdf`
[3] `http://www.geotools.org`
[4] `https://jena.apache.org/`
[5] `http://docs.openlinksw.com/virtuoso/rdfsparqlgeospat.html`

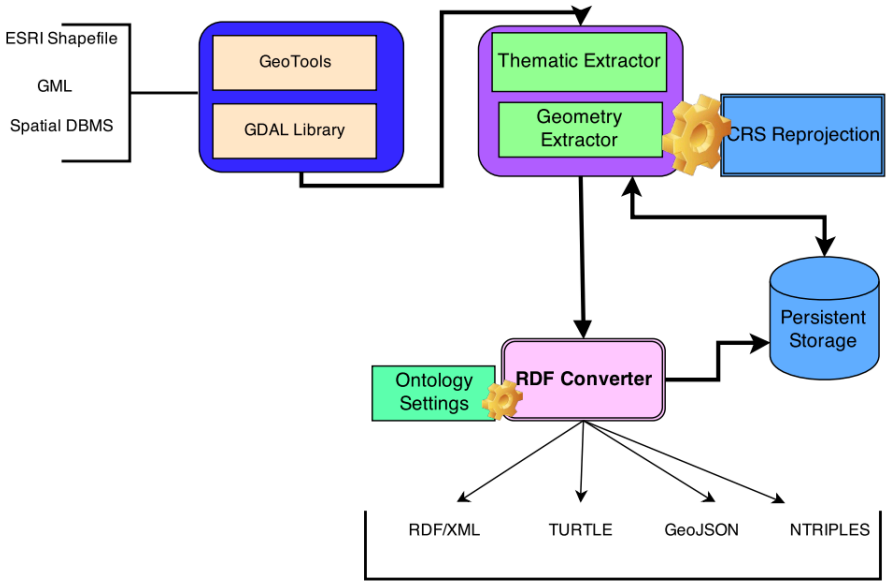


Figure 3.1: Generic architecture of tools for converting raw geospatial data into RDF.

3.3 GeomRDF: Datalift tool for Converting Geodata

GeomRDF [31] is a tool developed within the Datalift platform that transforms geospatial dataset from traditional GIS formats into RDF, and overcome the limitations of the existing tools mentioned in Section 3.2.4. GeoRDF is based on a vocabulary that reuses and extends GeoSPARQL and NeoGeo so that geometries can be defined in any CRS, and represented both as structured geometries and GeoSPARQL standard compliant. GeomRDF is composed of three components:

- **input parsers:** This component extracts from the different input format all the features and their descriptions.
- **feature parsers:** This component extracts for all the features their properties depending on their type, either thematic (attributes and properties of a geographic entity) or geometric (the geometry associated to the entity). For example, in the case of a Multipolygon, the parser stores first all the Polygons composing the original Multipolygon. Then, it stores the exterior and the eventual LinearRings for each Polygon, as well as the points included in the LinearRings. Finally the coordinates of each point are also stored. At this stage, GeoRDF provides a *"CRS reprojection"* functionality, which consists of transforming on-the-fly between different CRSs, before the storage of the geometries.
- **RDF Builder:** This module is in charge of generating RDF triples according to the `geom` vocabulary for geometry, **geofla** and **topo** for different topographic entities.

GeomRDF is implemented as a module of the RDF publication platform Datalift. Moreover, it has been validated against the French Administrative Units dataset. GeomRDF can also can also bee used as a stand-alone library and can be accessed at `https://github.com/fhamdi/GeomRDF`. Listing 3.1 presents a snippet in TURTLE of GeomRDF for geometry of the city of Nice (France). It also contains the structured modeling of a MultiPolygon as a set of Polygon, containing Points in an LinearRing.

3.4 Geodata Providers and Access

So far, the Web of data has taken advantage of geocoding technologies for publishing large amounts of data. For example, Geonames provides more than 10 millions records (e.g. $5,240,032$ resources of the form `http://sws.geonames.org/10000/`) while LinkedGeoData has more than $60,356,364$ triples. All the above mentioned data are diverse in their structure, the access point (SPARQL endpoint, web service or API), the entities they represent and the vocabularies used for describing them. Table 3.1 summarizes for different providers the number of geodata available (resources, triples) and how the data can be accessed.

Provider	**#Geodata**	**Data access**
DBpedia	727,232 triples	SPARQL endpoint
Geonames	5 240,032 (feature).	API
LinkedGeoData	60,356,364 triples	SPARQL endpoint, Snorql
Foursquare	N/A	API
Freebase	8,5MB	RDF Freebase Service
Ordnance Survey(Cities)	6,295 triples	Talis API
GeoLinkedData.es	101 018 triples	SPARQL endpoint
Google Places	N/A	Google API
GADM	682 605 triples	Web Service
NUTS	316 238 triples	Web Service
IGN experimental	629,716 triples	SPARQL endpoint
LOD Greek	634 KTriples	SPARQL endpoint

Table 3.1: Geodata by provider and their different access type

3.5 Scenario: 7^{th} Arrondissement of Paris

The 7^{th} arrondissement of Paris is one of the 20 arrondissements (administrative districts) of the capital city of France. It includes some of Paris's major tourist attractions such as the Eiffel Tower, some world famous museums (e.g: *musée d'Orsay*) and contains a number of French national institutions, including numerous govern-

```
@prefix geom:<http://data.ign.fr/def/geometrie#> .
@prefix rgeofla:<http://data.ign.fr/id/geofla/commune/> .
@orefix gsp: <http://www.opengis.net/ont/geosparql#> .

rgeofla:Multipolygon_11130 a geom:MultiPolygon ;
geom:crs <http://data.ign.fr/id/ignf/crs/WGS84GDD> ;
geom:polygonMember _:polygon_11130_1 ;
geom:polygonMember _:polygon_11130_2 .

_:polygon_11130_1 a geom:Polygon ;
geom:crs <http://data.ign.fr/id/ignf/crs/WGS84GDD> ;
geom:exterior  _:linearRing_11130_1 .

_:polygon_11130_2 a geom:Polygon ;
geom:crs <http://data.ign.fr/id/ignf/crs/WGS84GDD> ;
geom:exterior  _:linearRing_11130_2 .


_:linearRing_11130_1 a geom:LinearRing ;
geom:crs <http://data.ign.fr/id/ignf/crs/WGS84GDD> ;
geom:points  _:points_11130_1 ;
geom:firstAndLast _:point_11130_10 .


_:linearRing_11130_2 a geom:LinearRing ;
geom:crs <http://data.ign.fr/id/ignf/crs/WGS84GDD> ;
geom:points  _:points_11130_2 ;
geom:firstAndLast _:point_11130_2 .


_:points_11130_1 a geom:PointsList; geom:firstAndLast _:point_11130_10 ;
rdf:rest _:point_11130_11 .
_:point_11130_11  a geom:PointsList; rdf:first _:point_11130_12 ;
rdf:rest _:point_11130_13 .
.....
_:point_11130_129  a geom:PointsList; rdf:first _:point_11130_130 ;
rdf:rest _:point_11130_1 .


_:point_11130_10 a geom:Point ;
geom:crs <http://data.ign.fr/id/ignf/crs/WGS84GDD> ;
geom:coordX "7.308745254207776"^^xsd:double ;
geom:coordY "43.69237084089203"^^xsd:double .

rgeofla:06088  a geofla:Commune ;
rdfs:label "NICE"@fr ;
geom:geometry rgeofla:Multipolygon_11130 ;
gsp:asWKT "<http://data.ign.fr/id/ignf/crs/WGS84GDD> MULTIPOLYGON
    (((7.308745254207776 43.69237084089203, 7.306051040744396
    43.68445728916297, ....., 7.308745254207776 43.69237084089203)))"^^gsp
    :wktLiteral .
```

Listing 3.1: Sample of structured geometry of the city of Nice

ment ministries[6]. We use it throughout this paper to highlight the diversity of representations one can use for this geographical entity. We assume that this district should be modeled as a POLYGON composed of a number of POINTs needed to "interpolate" its effective boundaries. We assume the use of the WGS84[7] geodetic system.

3.5.1 DBpedia Modeling

We provide below an excerpt of the DBpedia description for the same resource:

```
dbpedia:7th_arrondissement_of_Paris a gml:_Feature ;
  a <http://dbpedia.org/class/yago/1900SummerOlympicVenuEs>
  rdfs:label "7. arrondissementti (Pariisi)"@fi; (14 different languages)
  dbpprop:commune "Paris" ;
  dbpprop:departement  dbpedia:Paris ;
  dbpprop:region dbpedia:Ile-de-France_(region) ;
  grs:point "48.85916666666667 2.312777777777778" ;
  geo:geometry "POINT(2.31278 48.8592)" ;
  geo:lat "48.859165"^^xsd:float;
  geo:long "2.312778"^^xsd:float.
```

First, we observe that the type `gml:_Feature` and the property `grs:point` are not resolvable since there are no OWL ontologies that provide a description of them. Second, the property `geo:geometry` used by DBpedia is not defined in the WGS84 vocabulary. For the geometry, the 7th arrondissement is a simple POINT defined by a latitude and a longitude.

3.5.2 Geonames Modeling

In Geonames, the 7^{th} arrondissement is considered as a 3^{rd} order administrative division, represented by a POINT for the geometry model. The RDF description of this resource gives other information such as the alternate name in French, the country code and the number of inhabitants.

```
gnr:6618613 a gn:Feature ;
  gn:name "Paris 07";
  gn:alternateName "7Ã¨me arrondissement";
  gn:featureClass gn:A [
    a skos:ConceptScheme ;
    rdfs:comment "country, state, region ..."@en .
  ] ;
  gn:featureCode gn:A.ADM4 [
    a skos:Concept ;
    rdfs:comment "a subdivision of a third-order administrative division"@en .
  ];
  gn:countryCode "FR";
  gn:population "57410";
```

[6] http://en.wikipedia.org/wiki/7th_arrondissement_of_Paris
[7] http://en.wikipedia.org/wiki/World_Geodetic_System

```
    geo:lat "48.8565";
    geo:long "2.321".
```

3.5.3 LinkedGeoData Modeling

In LinkedGeoData dataset, the district is a `lgdo:Suburb` which is subClass of `ldgo:Place`. Its geometry is still modeled as a POINT and not as a complex geometry of type POLYGON as we could have expected for this type of spatial object.

```
lgd:node248177663 a lgdo:Suburb ;
  rdfs:label "7th Arrondissement"@en , "7e Arrondissement" ;
  lgdo:contributor lgd:user13442 ;
  lgdo:ref%3AINSEE 75107 ;
  lgdp:alt_name "VIIe Arrondissement" ;
  georss:point "48.8570281 2.3201953" ;
  geo:lat 48.8570281 ;
  geo:long 2.3201953 .
```

3.5.4 Discussion

These samples from DBpedia, Geonames and LinkedGeoData give an overview of the different views of the same reality, in this case the district of the 7^{th} Arrondissement in Paris. Regarding the "symbolic representation", two datasets opted for "Feature" (DBpedia and Geonames) while LGD classifies it as a "Suburb" or "Place". They all represent the shape of the district as a POINT which is not very efficient if we consider a query such as *show all monuments located within the 7th arrondissement of international importance*. To address this type of query and more complicated ones, there is a need for more advanced modeling as we describe in the next section.

3.6 Survey on Triple Stores

A triple store is a back-end that has some form of persistent storage of RDF data and provides mechanisms to run SPARQL [32] queries against that data. The SPARQL support can either be built-in as part of the main tool, or an add-on installed separately. In this section, we discuss the most used triple stores based on the list maintained by sparqles [33] and we classify them in two group: (i) "generic triple stores" that handle any RDF triples and (ii) "geospatial triple stores", the ones designed to handle geographic data and implement topological functions.

3.6.1 Generic Triple Stores

We briefly describes some well-known triple stores that has proven to have passed the "10 Billion Statements", that is they are able to load and handle more than 10 billion of triples.

3.6.1.1 Virtuoso

Virtuoso is a triple store developed by OpenLink[8] and releases both in Open source and commercial versions. It implements part of SPARQL1.1 and store billion of triples. The geospatial extension implements subset of SQL MM functions. A geometry stores in Virtuoso uses a special RDF typed literal `virtrdf:Geometry`[9]. Such a literal is automatically indexed in an R tree index. Only geometry in WGS84 can be manipulated. Virtuoso is used as back-end for one of the most used dataset in the LOD cloud, DBPEDIA[10]

3.6.1.2 OWLIM

OWLIM is a semantic repository developed by Ontotext[11], which provides support and querying of two-dimensional point geometries modeled with the W3C Geo vocabulary. It implements spatial predicates represented as property funcions. The available four operations are: Buffer, Distance, Nearby and Point-in-polygon. OWLIM is implemented as a storage and inference layer for Sesamne, with custom spatial index.

3.6.1.3 AllegroGraph

AllegroGraph[12] is another RDF store developed and maintained by Franz Inc. which stores geospatial data types as native data structures. Support is provided both for Cartesian coordinate systems and for spherical coordinate systems. Every datum in an AllegroGraph store is a UPI, and for geospatial data, the added UPI type is `:geospatial` with type code `+geospatial+`. Geometries are assigned to geometric objects through the use of property
`<http://franz.com/ns/allegrograph/3.0/geospatial/pos>`, and for querying a GEO operator is introduced to express geospatial query patterns in SPARQL. AllegroGraph provides some operations on geodata, such as Bounding Box, Distance and Buffer.

3.6.2 Geospatial Triple Stores

We discuss in this section two triple stores built for indexing and querying geospatial data, Parliament and Strabon. We acknowledge that there might be some other solutions (e.g., Oracle DBMS version 111g, release 2). More details can be found in [14, 34, 35]. In [35], the authors present a benchmark of Geospatial RDF stores in the context of the TELIOS project[13], which uses real-world and synthetic data

[8] `http://virtuoso.openlinksw.com/`
[9] `http://www.openlinksw.com/schemas/virtrdf#`
[10] http://dbpedia.org/sparql
[11] `http://www.ontotext.com/owlim`
[12] `http://franz.com/agraph/allegrograph/`
[13] http://geographica.di.uoa.gr

to test the offered functionality and the performance of some prominent geospatial RDF stores.

3.6.2.1 Parliament

The RDF store Parliament[14] developed by BBN Technologies fully implements most of the functionality of GeoSPARQL [34]. Parliament has been extended to provide the first implementation of GeoSPARQL, therefore geometries are represented using the WKT and GML serializations. Therefore, topological functions belonging in the OGC Simple Feature Access, Egenhofer and RCC8 families are exposed by Parliament. Topological properties and non-topological functions are also available.In addition, multiple coordinate reference systems may be used. Parliament is implemented in C++, and Java Native Interface is used to couple with Jena, and includes a rule engine, serving as a means of inference. It registers the presence of a spatial object in an R-Tree.

3.6.2.2 Strabon

Strabon[15] is a semantic spatio-temporal RDF store for `stRDF`[16] and `stSPARQL` [36]. Strabon extends the well-known RDF store Sesame [cite], allowing it to manage both thematic and spatial data expressed in `stRDF`. PostGIS is used as the relational backend of Strabon. stRDF uses OGC standards (the OGC SFA specification in particular) for the representation of geospatial data. The datatypes strdf:WKT and strdf:GML are introduced to represent geometries serialized using the OGC standards WKT and GML. Strabon allows geometries to be expressed in any coordinate reference system defined by the EPSG (European Petroleum Survey Group) or OGC. In addition, the latest version of Strabon implements the GeoSPARQL Core, Geometry extension and Geometry topology extension components.

3.6.3 How to choose a Triple Store

There are many criteria that can be used to assess triple stores based on the requirements dataset publishers. Most importantly are the compatibility with the set of standards for RDF and SPARQL (e.g., SPAQRL1.0, SPARQL1.1). Table 3.2 provides an overview comparing "generic" triple stores. Currently, none of the triple stores listed supports security mechanisms built natively.
Regarding specific geospatial triple stores (or extensions of geospatial in RDF stores), more specific requirements are needed to index and provide native functions/operations over geometries. In Table 3.3, triple stores are compared based on (i) the types of geometries supported, (ii) the coverage of spatial functions, (iii) the compliance with GeoSPARQL standard, (iv) the extensions to existing vocabularies and SPARQL to manage geodata.

[14] `http://parliament.semwebcentral.org`
[15] `http://www.strabon.di.uoa.gr`
[16] `http://strdf.di.uoa.gr/ontology`

Serialization and Triple stores: We also advocate the use of properties that can provide compatibility with other formats (GML, KML, etc.). This choice can be triple store independent, as there could be ways to use content-negotiation to reach the same result. In Table 3.3, `Open Sahara`, `Parliament` and `Virtuoso` are WKT/GML-compliant with respectively 23 and 13 functions dealing with geodata. Moreover, the choice of the triple store (e.g.,Virtuoso[17] vs Open Sahara) is not really an issue, as the IndexingSail[18] service could also be wrapped on-top of Virtuoso to support full OpenGIS Simple Features functions[19].

Table 3.2: Survey of some generic popular triple stores.

Triplestore **Reasoning**	**SPARQL1.0** **10 billion statement**	**SPARQL1.1** **Clustering**	**SPARQL Update** **Open source**
`Virtuoso`	Yes	Partial	Non-std
Rules	Yes	Yes	Yes/No
`OWLIM`	Yes	Yes	Yes
Rules	Yes	Yes	No
`AllegroGraph`	Yes	Partial	Yes
Rules	Yes	Yes	No
`4Store`	Yes	Partial	Yes
Add-on	Maybe	Yes	Yes
`Sesame`	Yes	Partial	Yes
Little	No	No	Yes
`Fuseki`	Yes	Yes	Yes
Rules	No	No	Yes

3.7 Datalift: A tool for Managing Linked (Geo)Data Publishing Workflow

In this section, we present the Datalift platform, as a tool to help for lifting raw data to RDF. existing tools that manage the workflow of publishing geodata on the Web. To the best of our knowledge, a related framework providing similar functionalities is the GeoKnow Stack that we discuss in section 3.7.2.

3.7.1 Datalift Platform

Datalift is an open source platform [37] helping to lift raw data sources or legacy data to semantic interlinked data sources. The ambition of DataLift is to act as a catalyst for the emergence of the Web of Data by providing a complete path from

[17]Here we used Virtuoso Open Edition, V6.xx

[18]`https://dev.opensahara.com/projects/useekm/wiki/IndexingSail`

[19]`http://www.opengeospatial.org/standards/sfs`

Table 3.3: Triple stores survey with respect to geometry types supported and geospatial functions implemented.

Triplestore **Geometry supported**	**WKT-Compliance** **Geospatial Functions**	**GML-Compliance** **Geovocabulary**
`Virtuoso` Point	Yes SQL/MM (subset)	Yes W3C Geo + Typed Literal
`AllegroGraph` Point	- Buffer, Bounding Box, Distance	– "strip" mapping data
`OWLIM-SE` Point	N/A Distance, Buffer, Nearby, Within	N/A W3C Geo
`Open Sahara` Point, Line, Polygons	Yes OGC-SFA, Egenhofer, RCC-8	Yes Typed Literal
`Parliament` Point, Line, Polygons	Yes OGC-SFA, Egenhofer, RCC-8	Yes GeoSPARQL vocabulary
`Strabon` Point, Line, Polygons	Yes OGC-SFA, Egenhofer, RCC-8	Yes stRDF

raw data to fully interlinked, identified, and qualified linked datasets. The Datalift platform supports the following stages in lifting the data:

1. Selection of ontologies for publishing data;
2. Conversion of data to the appropriate format (e.g., from CSV to RDF);
3. Interlinking of data with other data sources;
4. Publication of linked data ;
5. Access control and license management.

Figure 3.2 gives an overview of the different steps in lifting raw source data into RDF using different modules of Datalift.

3.7.1.1 Functionalities of the Datalift platform

The architecture of Datalift is modular. Several levels of abstraction allow decoupling between the different stages from raw data to semantic data. The dataset selection allows us to identify the data to be published and migrate them to a first RDF version. The ontologies selection step asks the user to input a set of vocabularies', terms that will be used to describe the lifted data. Once the terms are selected, they can be mapped to the raw RDF and then converted to properly formatted RDF. The data is then published on the DataLift SPARQL endpoint. Finally, the process aims at providing links from the newly published data to other datasets already published as Linked Data on the Web. Figure 3.2 corresponds to a visual

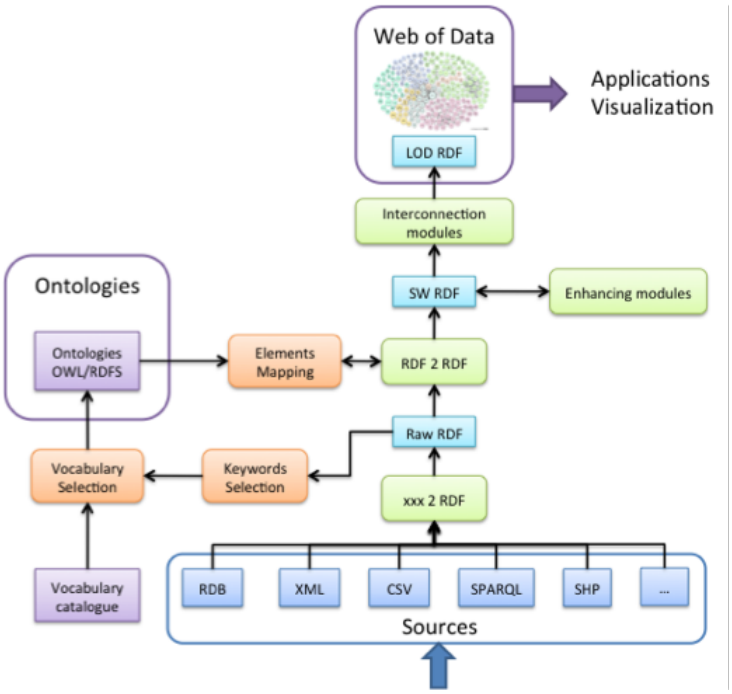


Figure 3.2: Lifting process of raw data source into RDF using Datalift Platform

depiction of the workflow to convert raw data into "structured" RDF data. Figure 3.7.1.1 depicts the architecture of Datalift, consisting of different modules for:

1. **Dataset Selection** The first step of the data lifting process is to identify and access the datasets to be processed. A dataset is either a file or the result of a query to retrieve data from a datastore. The kinds of files currently considered are CSV, RDF, XML, GML and Shape files. Queries are SQL queries sent to an RDBMS or SPARQL queries on a triple store.

2. **Ontologies Selection:** The publisher of a dataset should be able to select the vocabularies that are the most suitable to describe the data, and the least possible terms should be created specifically for a dataset publication task. The Linked Open Vocabularies [38] (LOV) developed in Datalift provides easy access methods to this ecosystem of vocabularies, and in particular by making explicit the ways they link to each other and providing metrics on how they are used in the linked data cloud. LOV is integrated as module in the DataLift platform to assist the ontology selection.

3. **Data Conversion:** Once URIs are created and a set of vocabulary terms able to represent the data is selected, it is time to convert the source dataset into a more precise RDF representation. Many tools exist to convert various structured data sources to RDF. The major source of structured data on the Web comes from spreadsheets, relational databases and XML files. Two steps are provided. First, a conversion from the source format to raw RDF is performed.

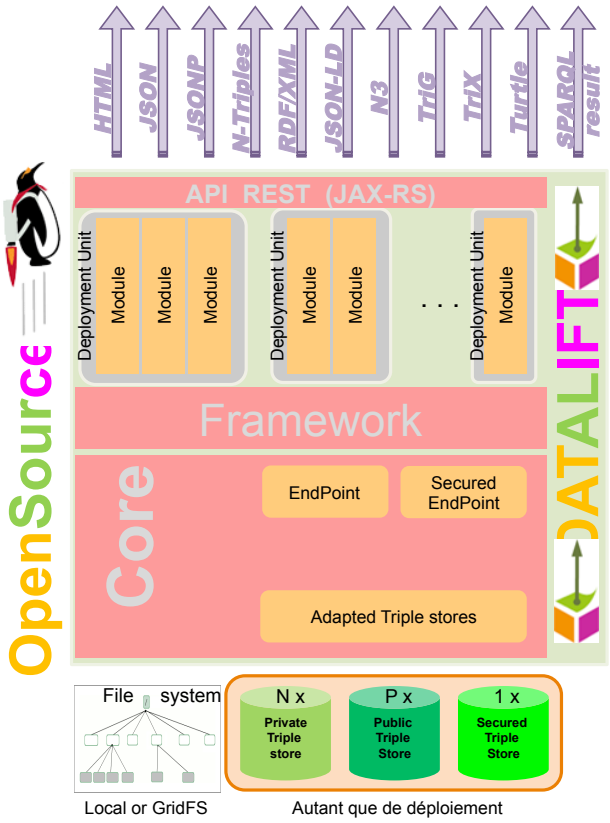


Figure 3.3: Architecture of Datalift platform.

Second, a conversion of the raw RDF into "well-formed" RDF using selected vocabularies is performed using SPARQL Construct queries. Most tools provide spreadsheet conversion to CSV, and CSV to RDF is straightforward, each line becoming a resource, and columns becoming RDF properties. The W3C RDB2RDF WG[20] proposes the Direct Mapping to automatically generate RDF from the tables but without using any vocabulary, and R2RML[21] to assign vocabulary terms to the database schema. In the case of XML, a generic XSLT transformation is performed to produce RDF from a wide range of XML documents. The DataLift platform provides a graphical interface to help mapping the data to selected vocabulary terms.

4. **Data Protection:** This module is linked to Apache Shiro for obtaining the information, i.e., username and password, about the user who is accessing the platform. The module[22] checks which data is targeted by the user's query and then verifies whether the user can access the requested data. This verification leads to three kinds of possible answers, depending on the access privileges

[20] http://www.w3.org/2001/sw/rdb2rdf/
[21] http://www.w3.org/TR/r2rml/
[22] http://wimmics.inria.fr/projects/shi3ld/

of the user: some of the requested data is returned, all the requested data is returned, or no data is returned. This means that the user's query is filtered in such a way that she is allowed to access only the data she is granted access to. The access policies are expressed using RDF and SPARQL 1.1[ref-sparql11] Semantic Web languages thus provide a completely standard way of expressing and enforcing access control rules.

5. **Data Interlinking:** The interlinking step provides means to link datasets published through the Datalift platform with other datasets available on the Web of Data. Technically, the module helps to find equivalence links in the form of "owl:sameAs" relations. An analysis of the vocabulary terms used by the published data set and a potential data set to be interlinked is performed. When the vocabulary terms are different, the module checks if alignments between the terms used by the two data sets are available.Here the alignment server provided with the Alignment API[23] is used for that purpose. The correspondences are translated into SPARQL graph patterns and transformation functions are combined into a SILK script.

6. **Data Publication:** This module aims at publishing the data obtained from the previous steps to a triple store, either public or private. The providers can restrict which graphs can be accessible, they could decide whether to provide just a "Linked Data" or a "Linked Open Data". Datalift comes by default with Sesame , but provides API for connecting to Allegrograph, OWLIM, and Virtuoso triple stores as well.

Installation: All the documentation for installing Datalift is available at `http://datalift.org/wiki/index.php/Platform_installation_(english)`. The latest version of the platform is announced at `http://datalift.org/en/node/24`, which is still a work in progress until the mature and stable version is launched and deployed.

Usage: The data lifting workflow has several distinct steps. DataLift makes it possible to replay each step in producing different results for each step. To facilitate access to all the different treatments and their results, they are grouped as one project. The project gathers together the various sources used and the results of all treatments done. Each module has its own way to be used within the lifting process in DataLift. For more details, the readers are encouraged to read this resource at `http://datalift.org/wiki/index.php/How_to_use_the_Datalift_platform_to_publish_a_dataset_on_the_Web#The_lifting_project`.

[23] `http://alignapi.gforge.inria.fr/`

3.7.2 Related Work: GeoKnow Stack

The GeoKnow Stack is a workbench developed within the Geoknow project[24], aiming at bringing geospatial knowledge integration to the Linked Data, with reasoning on Billion-triples and providing data provenance, and adaptive authoring, exploration and curation of geospatial data. GeoKnow Stack consists of eight tools integrated in 6 modules for Extraction, storage and querying, authoring, linking, enrichment and exploration. Figure 3.7.2 shows the architecture.

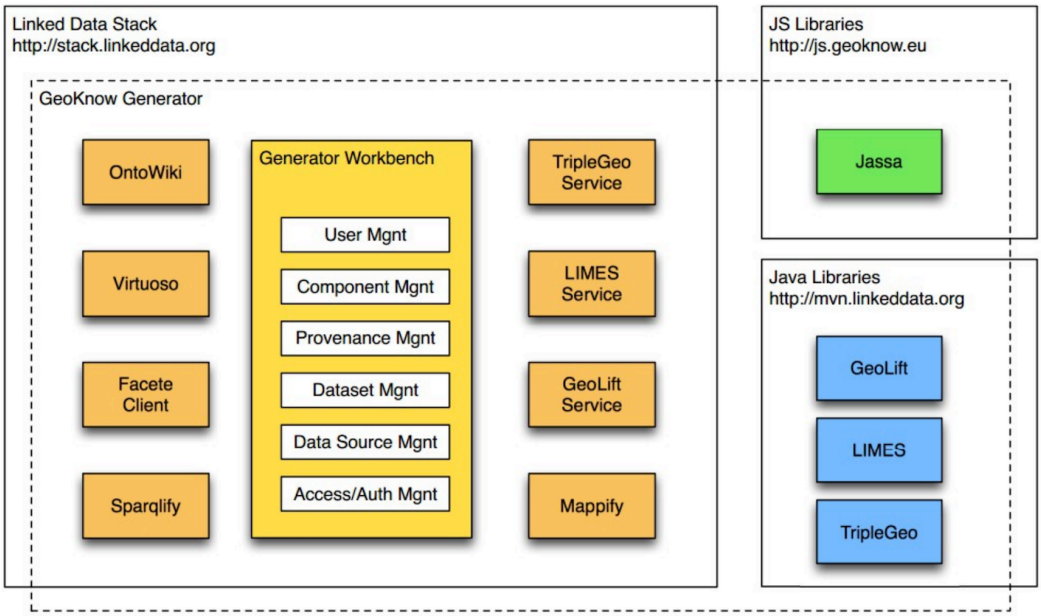


Figure 3.4: Architecture of the Geoknow Stack.

Below a brief description of the main modules:

- **Extraction and Loading:** The module is in charge of loading/importing RDF datasets, extract and convert legacy datasets using extractors/mappers such as TripleGeo (Cf.Section 3.2) and Sparqlify[25]. Sparqlify is a SPARQL-SQL rewriter that enables one to define RDF views on relational databases and query them with SPARQL. Currently in alpha state, it powers the Linked-Data Interface of the LinkedGeoData Server â i.e. it provides access to billions of virtual triples from the OpenStreetMap database.

- **Storage and Querying:** The module is powered by Virtuoso 7.0 triple store for querying the datasets.

- **Authoring module** integrated the OntoWiki tool [39] that facilitates the visual presentation of a knowledge base as an information map, with different

[24] http://geoknow.eu/
[25] http://sparqlify.org/

views on instance data. It enables intuitive authoring of semantic content, with an inline WYSIWYG editing mode for editing RDF content.

- **Linking and Fusion:** To achieve this module, LIMES is the tool integrated in the workbench. LIMES is an abbreviation of the LInk discovery framework for MEtric Spaces, a tool for interlinking resources on the Web of Data[26]. It implements time-efficient approaches for large-scale link discovery based on the characteristics of metric spaces [40]. It is easily configurable via a web interface or can also be downloaded as standalone tool for carrying out link discovery locally. LIMES can reduce the number of comparisons needed during the mapping process by several orders of magnitude [41]. The approaches implemented in LIMES include the original LIMES algorithm for edit distances, REEDED for weighted edit distances, HR3, HYPPO, and ORCHID. The algorithms implemented include the supervised, active and unsupervised versions of EAGLE, COALA and EUCLID [42].

- **Enrichment:** During the enrichment step, the tool GeoLift [43] to enrich geographic content using three techniques: linking, dereferencing and Natural Language Processing (NLP). The code is available at `https://github.com/AKSW/GeoLift/`.

- **Exploration:** The workflow proposes two tools for this module: Mappify and Facet. The former is a tool for exploring (geographical) Linked Data datasets on the Web, while the latter allows to explore the specific slice of data named "facet" of a Linked Data endpoint in a graphical way, by defining set of constraints on properties of the database. Once the facet is define, the information in the facet can be clicked-through in a tabular interface and visualized on a map. Facete is available at `http://144.76.166.111/facete/`.

3.7.3 Comparison between Geoknow Stack and Datalift

As described above, Geoknow Stack share similar goals and functionalities with Datalift. However, while the latter is cross-domain and generic, the former is more target for geospatial data. Moreover, Datalift implement connectors for many triple stores, while Geoknow Stack is powered by Virtuoso. Datalift is available in different platforms (Linux, Windows and Mac OS), while the current version of Geoknow can be installed only in Linux. Table 3.4 gives more dimensions of both frameworks.

3.8 Publishing French Administrative Units (GeoFla)

As a dataset dedicated to administrative units, GEOFLA®is very likely to be reused by other datasets, either by reusing directly its URIs for georeferencing needs, or by reusing its description of administrative units - labels, properties and geometries - for interlinking purposes.

[26] `http://aksw.org/Projects/LIMES.html`

Table 3.4: Comparison of Datalift with GeoKnow Stack

Features	Geoknow Stack	Datalift
Scope	Geospatial data	Cross-domain/Generic
Triple Store	Virtuoso 7	Sesame, Virtuoso 6, AllegroGraph
Shape2RDF	Few models for geometry	More generic approach
Interlinking Tool	LIMES	SILK
Publication	Publish directly in a graph	Export dump data in CSV, Turtle, NTriples
Access Control	N/A	Security Access Module
Installation	Expert level	One-click
Platform	Linux	Multi-platform
Deployment Environment	N/A	IGN, INSEE
Provenance	Partially (user)	Tracking PROV for each project
Visualization	Facete, Mappify	Sgvizler, RDFViz
Access Control	N/A	Security Access Module
Authoring	Wiki integrated	N/A

3.8.1 Data conversion

Geofla is delivered as a set of 4 shapefiles that describe the boundaries and properties of administrative units of mainland France (for CRS reasons, overseas territories are delivered within different shapefiles) : communes, cantons, arrondissements and departements. For the sake of our application, we have generated another shapefile describing regions by aggregating the geometries of the instances of departments based on their region's foreign key value. This dataset is updated every year. Publishing this data in RDF with unique identifiers on the Web will ease the interlinking with some existing datasets describing French boundaries in the wild. We follow a two steps conversion: we use the SHP2RDF module of Datalift to obtain a raw RDF from shapefiles, and the RDF2RDF module of Datalift using a set of SPARQL construct queries[27] for getting a refined RDF datasets using suitable vocabularies.

3.8.2 URI design policy

One of the requirements to publish data is to have unique ids and stable URIs[28] . Since our legacy databases have unique IDs to refer to the objects, we had to make sure they were unique at Web level. Thus, the base scheme for vocabularies

[27] https://github.com/gatemezing/ign-iswc2014/tree/master/rdf2rdf
[28] http://www.w3.org/TR/ld-bp/

URIs is: `http://data.ign.fr/def/`. Besides, the base schema for identifying a real world resource uses `http://{BASE}/id/`. For example, IGN main buildings are located in the commune with the URI `rgeofla:commune/94067`, corresponding to Saint-Mandé, and `rgeofla:departement/94` corresponds to the department "Val de Marne" to which the commune belongs. We had to make choices based on the set of best practices related to URI design[29] which should guarantee stable and human readable identifiers.

3.8.3 Interlinking with existing GeoData

We interlinked our datasets with NUTS, DBpedia FR[30] and GADM datasets. SILK [44] is used to interlink the departments in our dataset with departments in DBpedia FR, using labels and INSEE Code. We obtained 93 matches (all correct) while three are missing for the departments 07, 09 and 75[31]. The LIMES tool[32] is then used to perform the rest of the interlinking tasks [42] with the trigrams function based on the labels with restriction to France.

- Geofla-RDF with DBpedia FR: **23 252** links obtained. This results show the missing of nearly 13 435 communes not correctly typed in DBpedia FR as `Spatial Feature` or `Place`, or not having a French Wikipedia entry.

- Geofla-RDF with GADM (8 314 443 features): **70** links obtained: 10 communes, 51 departments and 9 regions. The property `gadm:in_country` is used to restrict the interlinking to France. E.g.: The city of Saint-Alban in Quebec is a commune in France.

- Geofla-RDF with NUTS (316 236 triples): Using a *"naive"* script with `trigrams` function on `geofla:Commune/rdfs:label` and `spatial:Feature/ramon:name` reveal two odd results located in Germany and Switzerland. The latter being the *JURA* and the former named *"Celle"*. In order to remove those odd effects, we add another restrictions based on ramon:code by filtering the ones located in France (136 features) . The final matchings give a total of **105** correct links: 14 communes, 75 departments and 16 regions.

The above results show good precision of the matching algorithm (score above 0.98) and a rather low recall value with DBPedia-FR (0.627). The few number of matched entities is likely due to the low coverage of French features in the datasets.
The SPARQL endpoint for the French Administrative dataset is available for querying at `http://data.ign.fr/id/sparql`.

[29] `http://www.w3.org/TR/ld-bp/#HTTP-URIS`
[30] `http://fr.dbpedia.org/`
[31] `https://github.com/gatemezing/ign-iswc2014/tree/master/interlinking/matched`
[32] `https://github.com/AKSW/LIMES.`

Datasets	Precision	Recall	F1-score
NUTS	0.98	1	0.90
GADM	1	0.86	0.92
DBpedia-FR	1	0.627	0.77

Table 3.5: Evaluation results in the interlinking process.

3.9 Publishing French Gazetteer

In this section, we present some first tests of converting BDTOPO® into RDF and interlinking with LinkedGeoData using LIMES. The results confirm the need for geographic publishers to publish georeference data on the Web.

Data conversion, URIs and Interlinking: Shapefiles are converted into RDF using the same two conversion process as for GEOFLA®. The URIs for each resource follow the pattern: `rtopo:CLASS/ID` for the feature, while `rtopo:geom/CLASS/ID` is used to reference the geometry of the resource. The gazetteer dataset in RDF is part of BD TOPO® database consisting of 1,137,543 triples (103,413 features). We chose LinkedGeoData (LGD) [33] to perform the alignments using the main class `lgdo:Amenity`[34] (5,543 001 triples), as they are closed to the features contained in the gazetteer. We perform the interlinking on the geometries using the `hausdorff` metric of LIMES tool. A total of **654** alignments was obtained above the threshold (0.9). This relatively low number of hits can be explained by the coverage of French data in LGD, and the subset of BDTOPO® used for the interlinking. Table 3.6 provides details of the alignments with subclasses of Amenity.

LGD Class	#links matched
lgdo:Shop	252
lgdo:TourismThing	30
lgdo:Craft	3
lgdo:AerowayThing	37
lgdo:AerialwayThing	11
lgdo:EmergencyThing	56
lgdo:HistoricThing	257
lgdo:MilitaryThing	8

Table 3.6: Interlinking results using the Hausdorff metric of LIMES tool between LinkedGeoData and toponyms in the French Gazetteer

[33] `http://linkedgeodata.org/sparql`
[34] `http://linkedgeodata.org/ontology/`

3.10 Publishing Addresses of OSM-France in RDF

OpenStreetMap France is working on providing the location addresses of France in different formats: CSV, ShapeFiles in a collaborative and open source fashion. The BANO project contains already 15 millions of indirect georeferencing locations. The geometries are POINTs and use WGS84 CRS. One of the requirement is to provide a RDF-ize version of the data, enriching the dataset with external/existing relevant ones. The goal of BANO2RDF with three basic requirements:

- **Requirement 1 (Req1)-Model:** Model the existing data according to an existing vocabulary, generic enough to cover the scope of the existing dataset.
- **Requirement 2 (Req2)-Provenance:** Add relevance metadata on the published dataset such as provenance, spatial coverage, licensing, authorship, etc.
- **Requirement 3 (Req3)-Stable URIs** : Define a policy to provide and ensure stable URIs for identifying uniquely each address entity on the Web.
- **Requirement 4 (Req4)-Interconnection:** Find relevant dataset already published on the Web to which interconnect for better discoverability.
- **Requirement 5 (Req5)-Access to data:** Provide primarily a frequent dump of the dataset in RDF.
- **Requirement 6 (Req6)-GeoSPARQL interoperability:** Provide different representations of geometries of locations that are interoperable with GeoSPARQL standard.

Based on the above requirements, the Location Core Vocabulary [45] is used in an *ad-hoc* script[35] to convert CSV data into RDF. URIs used to identify objects are of the form: `http://id.osmfr.org/bano/INSEE-CODE+FANTOIR-CODE+Street-Number`. For example, the location corresponding to EURECOM in SophiaTech (*450, route des chappes, biot, 06410 Biot, FRANCE*) is identified by the URI `<http://id.osmfr.org/bano/060180238L-450>`. Moreover, the property `locn:location` links to French Statistic dataset[36] for communes in France. Metadata are inserted at the beginning of the dataset in RDF corresponding to a department, using vocabularies to model license, spatial coverage of the data (i.e., `dcat`, `dcterms`, `foaf`). The extract below represents the metadata for the department of "ALPES-MARITIMES" which the commune of Biot belongs:

```
<http://www.openstreetmap.fr/bano/data/> a dcat:Catalog ;
dcterms:title "Donnees des adresses du projet BANO en RDF"@fr ;
dcterms:description "Le projet BANO en RDF de Base d'Adresses Nationale
Ouverte par OpenStreetMap France."@fr ;
foaf:homepage <http://openstreetmap.fr/bano> ;
```

[35] `https://github.com/osm-fr/bano/blob/master/out/csv2ttl.py`
[36] `http://id.insee.fr`

```
dcterms:language "fr" ;
dcterms:license <http://www.opendatacommons.org/licenses/odbl/> ;
dcterms:publisher <http://www.openstreetmap.fr/> ;
dcterms:issued "2014-05-14"^^xsd:date ; # data issued
dcterms:modified "2014-08-21"^^xsd:date ; #last modification
dcterms:spatial <http://id.insee.fr/geo/departement/06>,
 <http://id.insee.fr/geo/pays/france> .
```

Geometries are provided in three different representations: W3C WGS84, typed literal in WKT and geo URI; all using the property `loc:geometry`. Currently, the dataset is mostly available as dump files at `http://www.openstreetmap.fr/bano/data/`. However, an experimental endpoint has been set up to query at `http://eventmedia.eurecom.fr/sparql` with the named graph `http://data.osm.fr/bano/`, consisting of nearly 170 million triples.
Regarding the interconnection with more existing geodata, first results of mappings using LIMES for amenities in three big cities of France (Paris, Lyon and Marseille) (cf. Table 3.7) shows the low presence of the addresses for French territory. All the results of the mappings are available at `https://github.com/gatemezing/bano2rdf-matching`. Shops and restaurants are the resources with most `owl:sameAs` links between Bano2RDF and LGD datasets.

3.11 Status of French LOD cloud (FrLOD)

We summarize in this section our contributions for a *French LOD (FrLOD)* cloud, consisting of the different datasets published in RDF based on best practices for publishing data in 4-5 stars on the Web. Datasets published belong to different domains: geographical, governmental, statistical and educational. A `void` description of the FrLOD gives details of the different datasets published[37], links and applications/visualizations built on top of them. Figure 3.5 depicts a static diagram of the FrLOD, while in Table 3.8 gives an overview of the datasets, number of triples and domains. Altogether, the FrLOD represents 340 million RDF triples, which is nearly **10%** of the DBpedia 2014 release[38].

3.12 Spatial Queries

We illustrate in this section some queries making use of geospatial data and geometries functions implemented in endpoints. In the different queries, we use the following POINTs in WGS84: Eurecom is located at (7.0463, 43.6266) and Eiffel Tower at (2.2942 48.8628).

[37] An interactive version can be accessed at `http://www.eurecom.fr/~atemezin/work/frenchLOD.svg`

[38] `http://wiki.dbpedia.org/Datasets`

LGData Amenities	Bano-750xx (248,052)		Bano-130xx (401,404)		Bano-690xx (89,061)	
	#matched	%	#matched	%	#matched	%
Building (22,283)	05	0.022	12	0.053	0	0
Parking (250,516)	735	0.293	625	0.24	210	0.083
Shop (778,680)	21,171	2.71	8,556	1.098	3,049	0.391
School (318,287)	883	0.277	411	0.129	197	0.061
PlaceOfWorship (357,445)	272	0.076	193	0.053	31	0.008
Restaurant (260,675)	13,567	5.204	2,654	1.018	1,882	0.721
PublicBuilding (26,735)	97	0.362	64	0.239	21	0.078
PostOffice (87,731)	971	1.106	555	0.632	173	0.197

Table 3.7: Initial mappings of Bano2RDF with LGD amenities resources respectively in Paris, Marseille and Lyon. The links are obtained using LIMES tool with a threshold of .97 using the `Hausdorff` distance.

Dataset	**NbTriples**	**Domain**
French Admin Unit	173,639,128	Geography
Schools in France	1,454,564	Statistics
Eurecom LD	34,651	Education
CRS dataset	23, 377	Government
Toulouse Amenities	963, 618	Government
Bano Project	163, 723, 230	Geography
French Post Offices	371, 381	Government
French District Evolution	150, 356	Geography

Table 3.8: Overview of the content of our contribution to the French LOD Cloud

3.12.1 UC: Querying LinkedGeodata

The query below finds public buildings 10 km around Eurecom with their corresponding distance from LinkedGeodata endpoint. In this query, we use the functions of `st_distance`, `st_intersects` and `st_point`. Time used to answer this query is **341 ms**, monitored by using the time command on `CURL -g <query>`. The result is listed in Table 3.9

```
1 Prefix lgd:<http://linkedgeodata.org/>
2 Prefix lgdo:<http://linkedgeodata.org/ontology/>
```

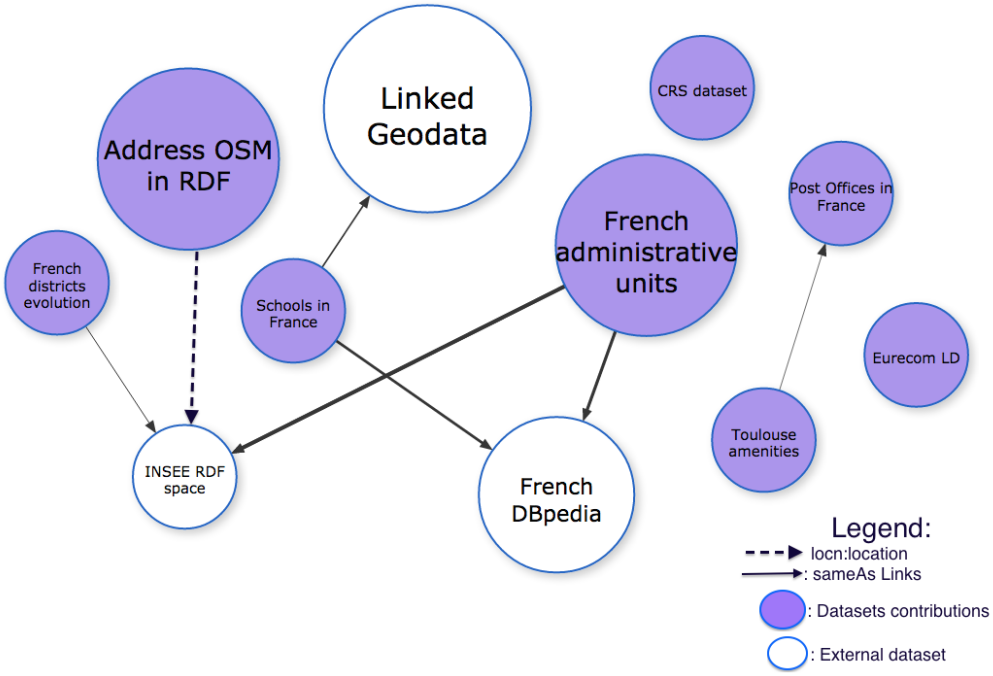


Figure 3.5: French LOD cloud diagram based on the different datasets published in 4-5 stars.

```
3
4 SELECT ?s ?name  (bif:st_distance(bif:st_point(?long, ?lat), bif:st_point
      (7.0463, 43.6266)) as ?distance)
5 WHERE
6  {
7        ?s a lgdo:PublicBuilding ; rdfs:label ?name .
8        ?s geo:lat ?lat ; geo:long ?long .
9      filter (bif:st_intersects(bif:st_point(?long, ?lat), bif:st_point
          (7.0463, 43.6266), 10 ))
10  }ORDER BY ASC (?distance)
```

Listing 3.2: Query on LinkedGeodata endpoint to find all public buildings 10 km around Eurecom building in SophiaTech.

3.12.2 UC: Querying FactForge (OWLIM)

The query below find in FactForge[39] the french departments within the Bounding Box of Eurecom and Eiffel Tower. We use the function `omgeo:within`. The results[40] consist of 36 departments and the query takes **1.369 s** to be completed.

```
PREFIX omgeo: <http://www.ontotext.com/owlim/geo#>
```

[39] http://factforge.net/sparql
[40] `http://goo.gl/dLc45p`

s	distance
http://linkedgeodata.org/triplify/node1645286605	1.43949
http://linkedgeodata.org/triplify/node1274078114	4.12131
http://linkedgeodata.org/triplify/node1082377863	5.06445
http://linkedgeodata.org/triplify/node1082377868	5.14575
http://linkedgeodata.org/triplify/node960122458	5.54149
http://linkedgeodata.org/triplify/node471974902	7.07093
http://linkedgeodata.org/triplify/node2209129663	7.59737

Table 3.9: Results of the public buildings 10 km around EURECOM from LinkedGeoData endpoint

```
PREFIX geo-ont: <http://www.geonames.org/ontology#>
SELECT DISTINCT ?link ?m
WHERE {
 ?link geo-ont:name ?m.
  ?link geo-ont:featureCode geo-ont:A.ADM2 .
  ?link geo-ont:parentCountry dbpedia:France .
  ?link geo-pos:lat ?lat2 .
  ?link geo-pos:long ?long2 .
 ?link omgeo:within( 43.6266 2.2942 48.8628 7.0463 ) }.
```

3.12.3 Case of Structured geometries

We query data.ign.fr that contains structured geometries of French departments, powered by Datalift, using the default triple store Sesame. The query finds all the departments containing in the Bounding Box formed by EURECOM and Eiffel Tower. The results consisting of 94 departments is obtained after **23.496s** when launched against `data.ign.fr/id/sparql`. Not all the SPARQL1.1 property paths [41] are fully implemented in Virtuoso endpoints.

```
 SELECT DISTINCT ?name  WHERE {
?dep a geofla:Departement .
?dep rdfs:label ?name .
?dep geom:geometry/geom:centroid ?ct .
?ct geom:coordX ?long ; geom:coordY ?lat.
?dep geom:geometry/geom:polygonMember/geom:exterior/geom:points ?pl .
?pl rdf:type geom:PointsList .
?pl (rdf:rest*/rdf:first)|geom:firstAndLast ?pm .
?pm rdf:type geom:Point .
?pm geom:coordX ?x .
?pm geom:coordY ?y .
```

[41] http://www.w3.org/TR/sparql11-query/#propertypaths

```
FILTER ((?x > 2.2942 && ?x < 7.0463) &&(?y > 43.6266 && ?y < 48.8628)) .
}
```

3.12.4 Summary

In this chapter, we have surveyed tools for extracting and converting geospatial data in RDF. Then, we describe GeomRDF, a tool developed within the Datalift project which goes beyond the state-of-art by providing structured geometries and conform to GeoSPARQL. Moreover, we have described the limitation of existing data models by discussing some recommendations to publishers of geodata on storage aspects. Similarly, an extensive description of Datalift tool used to publish data on the Web has been provided, with special focus of our contribution to build the French LOD cloud with datasets in 4-5 stars according to Linked Data principles. We have finished by showing real use-cases of SPARQL queries on top of those generated geospatial datasets.

Part II

Generating Visualizations for Linked Data

Chapter 4

Survey on Visualization Tools and Applications

"I think we have consensus, RDF is something you don't show your end users."[1]
Phil Archer (W3C Data Activity Lead)

4.1 Introduction

According to [46] the main goal of information visualization is to translate abstract information into a visual form that provides new insight about that information, in a clearly and precise form. In the traditional information visualization field, data classification, either quantitative or categorical, is useful for the purpose of visualization, and can make differences between tools. For example, hierarchical faceted metadata is used to build a set of category hierarchies where each dimension is relevant to the collection for navigation. The resulting interface is known as faceted navigation, or guided navigation [47]. However, visualizing structured data in RDF by taking advantage of the underlying semantics is challenging both by for the publishers and the users. For one hand, publishers need to build nice visualizations on top of their 4-5 stars datasets. On the other hand, lay users don't need to understand the complexity of the semantic web stack in order to quickly get insights of the data. Thus, adapting visual tools for exploring RDF datasets can bridge the gap between the complexity of semantic Web and simplicity in information exploration. In this chapter, we survey tools for visualizing structured data (section 4.2) and RDF data section 4.3). We then provide a classification of the tools for creating applications in the context of LOD (cf. section 4.4), along with the way applications are describe on the Web. Section 4.6 describes Linked Data applications, followed by the relevant information to describe applications built on top of government open datasets (section 4.6.2). The chapter ends with a brief summary.

4.2 Tools for visualizing Structured Data

In this section, we describe also visualization tools that natively do not take as input RDF data for two reasons:

[1] `https://twitter.com/philarcher1/status/507856407127814145`

- those tools are relatively "popular" for analyzing data exposed by the government and agencies (most of them in XLS, CSV) as they quickly make it easy to the users to build chart maps and compare with other datasets. One widely application is in the data journalism where facts are analyzed by those tools without waiting for the semantic publication of the data
- Also these tools have many options for visualizing data and are not totally adapted in the Semantic Web community.

4.2.1 Choosel

`Choosel` [48] is built on top of GWT and the Google App Engine (the backend can be modified to run on any servlet container). The client-side framework facilitates the interaction with visualization components, which can be wrappers around third party components and toolkits such as the Simile Timeline, Protovis and FlexViz. Choosel can integrate components developed using different technologies such as Flash and JavaScript. It is possible to implement visualization components that are compatible with the Choosel visualization component API. These visualization components can then be used to take advantage of Choosel features such as management of view synchronization, management of selections, and support for hovering and details on demand.

4.2.2 Many Eyes

`Many Eyes` [49] is a website that provides means to visualize data such as numbers, text and geographic information. It provides a range of visualizations including unusual ones such as "treemaps" and "phrase trees". All the charts made in Many Eyes are interactive, so it is possible to change what data is shown and how it is displayed. Many Eyes is also an online community where users can create groups (such as "Ebola Crisis" or "Kobane War") to organize, share and discuss data visualizations. Users can also comment on visualizations made by others, which is a good way to improve their work. The authors claim that it is useful because it users can build quick and easily visualizations from their own data, with the possibility to share them. is quick and easy to make and share great looking and fun to use visualizations from your own data. Data input formats are XLS, Plain text and HTML. The output formats are PNG or embeddable. However, using Many Eyes make public your data and the visualizations created with it. The license is proprietary of IBM.

4.2.3 D3.js

`D3.js` [50] is a JavaScript library for manipulating documents based on data. D3 uses HTML, SVG and CSS. D3 combines powerful visualization components, plugins[2] and a data-driven approach to Document Object Model (DOM) manipulation.

[2] `https://github.com/d3/d3-plugins`

D3 solves problems of efficient manipulation of documents based on data. Thus, avoids proprietary representation and affords flexibility, exposing the full capabilities of web standards such as CSS3, HTML5 and SVG. D3 supports large datasets and dynamic behaviors for interaction and animation.

D3 intention is to replace gradually Protovis[3], which is another tool to build customs visualizations in the browser, created by the same authors and which is no longer under active development. Although D3 is built on many of the concepts in Protovis, it improves support for animation and interaction. The difference between D3 and Protovis is in the type of visualizations they enable and the method of implementation. While Protovis excels at concise, declarative representations of static scenes, D3 focuses on efficient transformations: scene changes. This makes animation, interaction, complex and dynamic visualizations much easier to implement in D3. Also, by adopting the browser's native representation (HTML & SVG), D3 better integrates with other web technologies, such as CSS3 and other developer tools .

4.2.4 Google Visualization API

The Google Visualization API[4] establishes two conventions to expose data and visualize it on the web: (1) a common interface to expose data on the web and (2) a common interface to provide data to visualizations [51]. Because the Google Visualization API provides a platform that can be used to create, share and reuse visualizations written by the developer community at large, it provides means to create reports and dashboards as well as possibility to analyze and display data through the wealth of available visualization applications. Many kinds of visualizations are available. Google Visualization accepts data in two different ways: a direct construction as well as a JSON literal object, instantiated via the object `google.visualization.DataTable`. In the latter, the structure of this JSON format is the convention that Google API data sources are expected to return. So, a `google.visualization.DataTable` can be created using the results of an AJAX response. It is possible to retrieve and visualize RDF data. As long as the URL retrieved returns Google Visualization JSON, you can create a DataTable and give it to the visual construct to `draw()`. The results of a SPARQL query can be converted to the Google Visualization JSON using an XSL like the one used at RPI for data.gov]. A sample performing these steps is presented in the Tetherless World Constellation, named `SparqlProxy`[5] . It performs these steps for a client with a single HTTP request. By providing the URL of a sparql endpoint to be queried (using service_uri), a query (using query or query-uri), and a specification for return format as Google Visualization JSON (using output=gvds).

[3] `http://mbostock.github.com/protovis/`

[4] `https://developers.google.com/chart/interactive/docs/reference`

[5] `http://data-gov.tw.rpi.edu/ws/sparqlproxy.php`

All the visualizations are based on the type of the columns/fields of the data. While this is normal for tabular data, it is not the case for data exploiting semantics. In Linked Data, vocabularies are used for modeling datasets in RDF, thus making it difficult to reuse directly those tools. There is a need to build more generic tools that exploits the semantics and reuse the visual tools aforementioned.

4.3 Tools for visualizing RDF Data

Regarding the tools for visualizing Linked Data, the paper [52] analyses in detail the current approaches used to browse and visualize Linked Data, by identifying requirements for users classify into two groups: tech-savvy and lay-users. As the authors extensively surveyed more generic Linked Data browsers, with text-based presentation and visualization options, they provide some recommendations according to the size of the data such as fine-grained analysis among others. However, they do not target their study on tools that can easily help building visual Semantic Web-based applications. However, our approach is to study the tools used to build innovative applications for detecting the components that could be reusable across different domain y/o scope.

4.3.1 Linked Data API

The Linked Data API (LDA) [cite], provides a configurable way to access RDF data using simple RESTful URIs that are translated into queries to a SPARQL endpoint. The API layer is intended to be deployed as a proxy in front of a SPARQL endpoint to support:(i) Generation of documents (information resources) for the publishing of Linked Data; (ii) Provision of sophisticated querying and data extraction features, without the need for end-users to write SPARQL queries and (iii) Delivery of multiple output formats from these APIs, including a simple serialization of RDF in JSON syntax.
ELDA[6] is a java implementation of the LDA by Epimorphics. Elda comes with some pre-built samples and documentation, which allows us to build the specification to leverage the connection between the back-end (data in the triple store) and the front-end (visualizations for the user). The API layer helps to associate URIs with processing logic that extract data from the SPARQL endpoint using one or more SPARQL queries and then serialize the results using the format requested by the client. A URI is used to identify a single resource whose properties are to be retrieved or to identify a set of resources, either through the structure of the URI or through query parameters.

4.3.2 Sgvizler

`Sgvizler` [53] is a javascript which renders the result of SPARQL SELECT queries into charts or html elements. The name and tool relies queries against SPARQL

[6] `http://www.epimorphics.com/web/tools/elda.html`

endpoints using visualizations based on Google Visualization API, SPARQLer , Snorql[7] and Spark[8]. All the major chart types offered by the Google Visualization API are supported by Sgvizler. The user inputs a SPARQL query which is sent to a designated SPARQL endpoint. The endpoint must return the results back in SPARQL Query Results XML Format or SPARQL Query Results in JSON format. Sgvizler parses the results into the JSON format that Google prefers and displays the chart using the Google Visualization API or a custom-made visualization or formatting function. Sgvizler needs, in addition to the Google Visualization API, the javascript framework jQuery to work. One of the drawback of Sgvizler that it is up to the user to test the query and embed it into the HTML page.

4.3.3 Facete

Facete [54] is an exploration tool for (geographical) Linked Data datasets on the Web. Also called "Semmap", the application allows the user to explore the specific slice of data named *'facet'* of a Linked Data endpoint in a graphical way, available at `http://144.76.166.111/facete/`. The facet is created by defining a set of constraints on properties in the database. Once the facet is defined, the information in the facet can be clicked-through in a tabular interface and visualized on a map. The user can choose a SPARQL endpoint and graph for listing the content and visualize the dataset. The application is divided in three main views:

1. Selection: A tree-based structure of the dataset. It shows all items' properties and sub-properties.

2. Data: shows a tabular representation of the data in the facet. All properties that have been marked with an arrow symbol in the facet tree are shown as columns. The columns contain the property values for every item according to the selected filter criteria.

3. Geographical: A map view showing a representation of the elements in the facet with geo-coordinates available.

4.3.4 VisualBox

Visualbox[9] is another tool that aims at facilitating the creation of visualizations by providing an editor for SPARQL queries and different visual tools to visualize the data. Visualbox is derived from LODSPeaKr [55] mainly based on the Model-View - Component (MVC) paradigm. A visualization is created in a Component consisting of one or more SPARQL queries (models), and usually one (but sometimes more) templates (Views). Visualbox is target to users that have at least some basic knowledge of SPARQL and an understanding of RDF, and runs the query on the

[7] `http://dbpedia.org/snorql/`
[8] `http://code.google.com/p/rdf-spark`
[9] `http://alangrafu.github.io/visualbox/`

server side. Visualbox uses Haanga[10], a template engine that provides a syntax for creating templates by defining markers in a document (usually a HTML page) of the form variable that later will be compiled and replaced by values taken from a data source. One of the drawback of Visualbox is the impossibility to extend with custom visualization nor enable third party filters. Currently, it implements visualization filters for D3.js (5), Google Maps, Google Charts(6) and TimeKnots library (TimeLine with events)[11].

4.3.5 Payola

Payola [56] is a web framework for analyzing and visualizing Linked Data, and enables users to build instances of Linked Data visualization Model (LDVM) pipelines [57]. LDVM is an adaptation of the Data State Reference Model (DSRM) proposed by Chi [58] applied to visualizing RDF and Linked Data. It extends DSRM with three additional concepts that are reusable software components:

- **Analyzers**: They take as input compatible datasets, and perform adapted SPARQL queries: hierarchical dataset, geocoordinates dataset, etc.
- **Visualization transformers**: They can be any software component that transform data between different formats or perfom aggregations for better visualization. They are generally SPARQL CONTRUCT queries, with the input signatures corresponding to the FROM clauses and their output data samples corresponding to the CONSTRUCT clauses.
- **Visualizers**: They consume RDF data and produce a visualization a user can interact with. They are visual tools libraries consuming data in RDF/JSON[12]

Basically a user builds different instances of LDVM based on the datasets used in the analyzers and transformers. Figure 4.3.5 depicts a sample of a LDVM pipeline applied to two different datasets publised as LOD.

4.4 Discussion

There are currently many projects aiming at visualizing (RDF) Linked Data. A survey by Dadzie and Rowe [59] concluded with the fact that many visualization tools are not easy to use by lay users. In [60], there is a recent review of some visualizations tools that can be summarized as follows:

- *Vocabulary based visualization tools:* these tools are built for specific vocabularies and that help in visualizing data modelled according to those vocabularies, such as CubeViz [61], FOAF explorer[13] and Map4rdf [62]. They aim at visualizing data modelled respectively with `dq,foaf` and `geo+scovo`.

[10] `http://haanga.net`
[11] `https://github.com/alangrafu/timeknots`
[12] `https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-json/index.html`
[13] `http://foaf-visualizer.gnu.org.ua/`

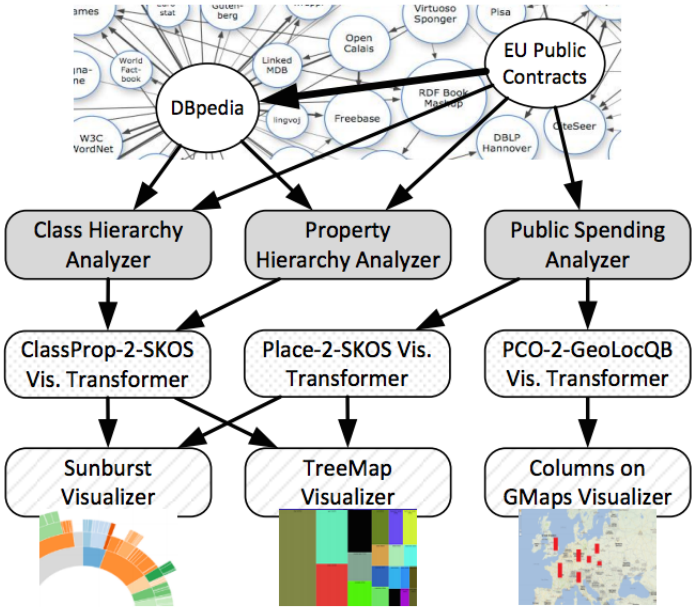


Figure 4.1: Sample application of analyzers and visualizers in a LDVM pipeline.

- *Mashup tools:* they are used to create mashup visualizations with different widgets and some data analysis, such as DERI Pipes [63]. Mashup tools can be integrated into the LD wizard to combine different visual views.

- *Generic RDF visualization tools:* they typically support data browsing and entity rendering. They can also be used to build applications. In this category, we can mention Graphity[14], lodlive[15] and Balloon Synopsis[16].

While these tools are often extensible and support specific domain datasets, they suffer from the following drawbacks:

- *They are not easy to set up and use by lay users.* Sometimes, users just need to have a visual summary of a dataset in order to start exploring the data. Our approach to this challenge is to provide such a lightweight javascript-based tool that supports a quick exploration task.

- *They do not make recommendation based on categories.* A tool similar to our approach is Facete[17][54] which shows a tree-based structure of a dataset based on some properties of an endpoint more target at geodata. A tabular view enables to visualize slices of data and a map view can be activated when there is geo data. Our approach aims to be more generic, offering more views

[14] `https://github.com/Graphity/graphity-browser`
[15] `http://en.lodlive.it/`
[16] `https://github.com/schlegel/balloon-synopsis`
[17] `http://cstadler.aksw.org/facete/`

(tabular, map, graph, charts, etc.) according to a systematic analysis of what are the high level categories present in a dataset.

The outcome of this state-of-the-art can then be used to assess different visual tools in the process of creating web-based visualizations. Some criteria can be used for assessing visual tools, such as (i) usability , (ii) visualization capabilities, (iii) data accessibility, (iv) deployment and (v) extensibility. In [64], the readers can find more details on this survey. Table 4.1 gives an overview of the selected tools studied based on the following features:

- *Data Formats* for the format of data taken as input by the tool;
- *Data Access*, for the way to access the data from the tool, such as web service, sparql endpoint, etc.
- *Language code*, the programming language used to develop the tool;
- *Type of Views*, the different views potentially accessible when using the tool;
- *Imported Libraries*, the external libraries available within the tool,
- *License* for the Intellectual Properties rights of the tool,
- *SemWeb Compliant*, whether the tool can be easily transposed or compliant with structured data; and

Table 4.1: Survey of some tools used for creating visualizations on the Web.

Tools **Views**	**Data Formats** **Libraries**	**Data Access** **License**	**Language** **SemWeb App**
`Choosel` Text/Map/Bar chart	XLS, CSV Time (Simile)/Protovis/Flexvis	API Open	GWT No
`Fresnel` Property/Labels	RDF Welkin/IsaViz/Haystack/CSS	– Open	RDF Yes
`Spark` Charts/Tabular	RDF-JSON –	SPARQL Open	PHP Yes
`LDA` –	RDF –	SPARQL Open	Java/PHP Yes
`SemWeb Import` Graph Node	RDF –	SPARQL CECILL-B	Netbeans Yes
`Many Eyes` charts/ trees/graphs/maps	XLS/Text/HTML –	API IBM	Java/Flash No
`D3.js` charts/ trees/graphs/maps	CSV/SVG, GeoJson Jquery/sizzle/colorbrewer	API Open	JavaScript Maybe
`Facet` Map,Facet view	RDF-JSON Jquery/ dynatree	SPARQL Open	JavaScript Yes
Sgvizler Map/Line chart, timeline/sparkline	RDF-JSON Google visualization API	SPARQL Open	JavaScript Yes
`Visual Box` Map/charts/ timeline/graphs	RDF Google charts/TimeKnots/D3.js	SPARQL Open	PHP/Django Yes
`Map4rdf` Facet/Map	RDF-JSON OSM Layers, Google Maps	SPARQL Open	Java/GWT Yes
`Exhibit` Map/Tile/Thumbnail/Tabular/Timeline	JSON Exhibit —	Data dump Open	JavaScript Yes
`Google Visualization API` Charts/ Charts/Maps/Dashboard	JSON/CSV AJAX API	API Open	JavaScript Possible

4.5 Describing Applications on the Web

4.5.1 Motivation

As many initiatives on Linked Open Data is growing, tools and technologies are getting more and more mature to help pro-consumers to leverage the lifting process of the data. At the same times, standardization bodies such as W3C are helping in providing best practices to publish Open Government Data by using appropriate vocabularies, taking care of stability in the URIs policies, and making links to other datasets. It is the case for instance of the Government Linked Data Working Group[18] which has released some best practices and vocabularies to help governments publishing their data using Semantic Web technologies. Having a look at different proposals of the Life Cycle of Government LD, one of the last stage is "Publication" where the data is released according to the 4-5 stars principles[19], with a given access to a SPARQL endpoint. However, for a better understanding of the data, one of the next step is usually to generate visualizations through intuitive visual tools(charts, graphs, etc.) that will benefit to citizens, data journalists and other public authorities to improve the quality of their decisions. Currently, one way of creating new applications is to look around previous initiatives to see what type of application exists already and make something similar according to a given dataset and domain. Another approach is by organizing *contests* where the challenges are to mash up unexpected datasets with clear and beautiful visualizations. Such approach is harder as developers also try figure out which tool and library is used for different applications. What if we describe applications according to the facets/views, datasets, visual tools used to build them? How are the types of information that can help create a vocabulary for annotating web-based visualizations online ?

4.5.2 Catalogs of Applications

We provide below two use case of the current description of applications developed by datasets published on the Web. We expose also the limitation of the approach as they don't fully make use of semantics for more discovery of visual tools, datasets used to developed such applications.

4.5.2.1 Open Data Service

The Open Data Service at the University of Southampton[20] has a register of all the applications developed using their datasets. A catalog of the Applications using the data is available at `http://id.southampton.ac.uk/dataset/apps`. Each application is described by giving three main categories of information:

[18] `http://www.w3.org/2011/gld/`
[19] `http://5stardata.info/`
[20] `http://data.southampton.ac.uk/apps.html`

Figure 4.2: Description of a Web application of an application at the Open Data Service

- The available distributions corresponding to the different formats HTML, RDF/XML and RDF/TURTLE ;
- Dataset information, which defines the type, the number of triples, license information, the publisher and the publication date.
- The provenance, such as files used to generate the dataset for building the application, as well as the script itself.

Currently, some vocabularies are used to model the catalog, such as `DCAT` vocabulary [65] and proprietary vocabularies. Each application is then described the type (Web, mobile Web, android, etc.), the authors, the date of creation and the datasets used to build the application. Figure 4.5.2.1 depicts the HTML view of a Web application for a searchable map for finding buildings within the University sites. This initiative seems to be isolated. Thus, there is a real need to have a common layer of semantics for describing such applications. This would benefits the interoperability and more discovery of applications on the Web.

4.5.2.2 RPI Applications

Another approach from the researchers at the Rensselaer Polytech Institute[21] is to put at the bottom of the static page of a demo/application showcasing the benefits of Open Data for `data.gov`[22] some basic metadata (description, URL to dataset, author), and also a link to the SPARQL query used for generating the application.

[21] `http://data-gov.tw.rpi.edu`
[22] `https://www.data.gov/`

As this information is human-readable and can help, the main drawback is the lack of a machine-readable version, using semantics to discover and connect different demos and datasets with authors. A more vocabulary can leverage the issue by annotating such applications to help discovering and aggregating other similar applications in other Open Data initiatives.

4.6 Linked Data Applications

According to [66], `Visualization` is *"the use of computer-supported, interactive visual representations to amplify cognition"*. So the unique object of visualization is developing insights from collected data. That justify why each time a new dataset is released, users always expected some showcases to play with the underlying datasets. It is true that many public open initiatives uses incentives actions like *challenges*, *datahack-day* or *contest*, etc. to find innovative applications that actually exhibit the benefits of datasets published. Visualizations play crucial role as they can easily find errors in a large collection; detect patterns in a dataset or help navigate through the dataset.

4.6.1 Typology of Applications

Jeni Tennison[23] defines in her blog[24] three categories of applications using online data:

- (i) *data-specific applications*, which are constructed around particular data sets that are known to the developer of the application; hence the visualizations obtained are of data-specific applications. Examples are the famous applications of *"Where does my money go"* in Greece[25] or UK[26]. Those applications are also called *"mashups"*.

- (ii) *vocabulary-specific applications*, which are constructed around particular vocabularies, wherever the data might be found that uses them. Examples here are FaceBook Social Graph API[27], IsaViz [67], among others.

- (iii) *generic applications*, which are constructed to navigate through any RDF that they find; e.g., Tabulator [68], OpenLink Data Explorer[28].

Because most mash-sups are data-specific applications, it is important and necessary to know what information the dataset contains. This could be achieved by giving the meaning of some properties or classes of the vocabularies used to create the dataset. Hence, what the data publisher needs to do very often is to make sure that the

[23] `http://www.theodi.org/people/jeni`
[24] `http://www.jenitennison.com/blog/node/126`
[25] `http://publicspending.medialab.ntua.gr/en/index.php`
[26] `http://wheredoesmymoneygo.org/`
[27] `https://developers.facebook.com/docs/plugins/`
[28] `http://ode.openlinksw.com/`

data they publish is documented. However, what is seeing in practice, is to consider using an intuitive visualization self-descriptive to both show the added-value of the data and its documentation.

Table 4.2: Gathering reusable information from openspending in Greece Application

Features	Value
Access Url	http://publicspending.medialab.ntua.gr/
Scope/Domain	Public spending, Government
Description	The application helps visualizing the most characteristic facts of the Greek public spending, interconnected to foreign expenditure and other data.
Supported Platform	Web
URL Policy	http://BASE/en/NAME-CHART.php e.g., http://{BASE}/en/toppayersday.php
Data Source	http://opendata.diavgeia.gov.fr; Greek Tax data (TAXIS)
Type of views	Bubble tree, column and bar charts
Visualization tools	HighchartsJS, Bubble TreeJS JqueryJS ; RaphaelJS
License	Open
Business Value	Not Commercial (Free)

4.6.2 On Reusable Applications

Many applications are built on top of datasets exposed in different open data governments initiatives. Generally, they are used to provide insight about the datasets and their usefulness. However, some of the applications could be generalized and reused if published adequately. Having some best practices in publishing applications on the Web could booster the interoperability between datasets and visual tools. To achieve this task, we first review some applications that have been developed on top of datasets opened by governments (UK, USA, France) and public local authorities. We made a random survey of thirteen (13) innovative applications [69] in various domains such as of security, health, finance, transportation, housing, city, foreign aid and education. Table 4.3 provides a summary of the surveyed applications; with names, types, countries and brief description.
The main template used in the survey was to gather the following information:

- the name of the application;
- the scope or the target domain of the application;
- a small and concise description;
- the platform on which the application can be deployed and view;

- the policy used for creating the URL of the application;
- the legacy data used to build the application, and a mention of the process of the lifting process of the raw data to RDF if available;
- the different views available of the application;
- comments or relevant drawback to mention;
- and the license of the application.

Table 4.2 provides the information extracted from `openspending in Greece` using the aforementioned template. Such information can be published using a vocabulary to help discover all the applications built on top of public spending data across different platforms.

Table 4.3: Some innovatives applications buit over Open Government Datasets

Application	**Domain**	**Type**	**Country**
`UK Crime`	Crimes	Web	UK
`UK Pharmacy`	Health, Pharmacy	Mobile/Android	UK
`Numberhood`	Local area dynamics	iPhone/iPad	UK
`BuSit London`	Public Transportation	Web and mobile	UK
`UK School Finder`	Education	Web	UK
`Where-can-I-Live`	House, transportations	Web	UK
`Opendatacommunities`	Local Government	Web	UK
`FlyOnTime`	Flights/airlines	Web	USA
`White House Visitor Search`	White House	Web	USA
`US-USAID/UK-DFID`	Foreign Aid	Web	USA
`Fourmisante`	Medicine/health-care	Web	France
`MaVilleVueDuCiel`	Local Government	Web	France
`Home'n'Go`	Housing	Web	France

4.7 Summary

In this chapter, we have described different tools used for visualizing data, structured and graph data. We have also discussed different types of applications currently built on top of government open data initiatives. The goal of this survey is to propose some new approaches of of generating and discovering visualizations and applications on the Web of Data.

Chapter 5

New Approaches for Generating Visualizations and Applications

"A Semantic Web application is one whose schema is expected to change."
(David Karger, MIT CSAIL)[1]

5.1 Introduction

The object of visualization is to develop insights from collected data. Moreover, according to Information Theory, vision is the sense that has the largest bandwidth (100 Mbits/s), which makes it the best suited channel to convey information to the brain [70]. Based on the Visual Information Seeking Mantra: *"overview first, zoom and filter, then details on demand"* [71], we advocate for more visual interactive representations of RDF graphs using SPARQL Endpoints. At the same time, we use the term "Linked Data Visualization", to refer to a *combination of charts, graphics, and other visual elements built on top of 4-5 stars datasets accessible via a SPARQL endpoint.* Linked Data offers some great advantages for publishing government data. The approach makes it easy to publish information in a way that allows it to be combined with other sets of data. The benefits also arise from the semantics associated to things, common identifiers for things, from the inherent extensibility of the RDF data model, and from the publication of data in a standard format. Linked Data is a great way of publishing information for diverse and distributed organizations, such as government [72].
Despite the presence of more and more datasets published as Linked Data, there is still a need to help end users to discover what (unknown) datasets describe by hiding the complexity of SPARQL queries from such users. The RDF model, its various serializations and the SPARQL query language are foreign to the majority of developers who understandably want to be able to use the tool chains that they are familiar with to access government data. Sometimes, publishing data purely as RDF, and providing access purely through SPARQL queries raises an unacceptable barrier onto the use of that data. Moreover, the task of identifying the key categories of datasets can help in selecting and matching the most suitable visualization types. In this chapter, we present our contribution on consuming datasets by generating applications target to lay-users. The remainder of this chapter is structured as follows. We first propose in section 5.2.2 some important categories that are worth

[1] A statement during his keynote at ESWC 2013 conference

visualizing and a set of mapping views associated with vocabularies (section 5.3). In Section 5.4, we describe the implementation of a wizard that can work on top of any RDF dataset. We detail the results of an experiment where high level categories and associated visualizations have been performed on numerous SPARQL endpoints (Section 5.8.2). Besides, we reverse engineer the GKP (Section 5.8) to look for the most important properties of an Entity. Then, we present two domain applications in events (Section 5.9) and statistics (Section 5.10). Finally, we discuss how to improve the discovery of applications developed in Open data events (Section 5.11) by proposing a vocabulary and a plugin to easily annotate their web pages and later generate RDF data.

5.2 Wizard for Visualizations

5.2.1 Background

With the growing adoption of the Linked Data principles, there is a real need to support data consumers in quickly getting visualizations that enable to explore a dataset. In order to involve more general Web users into the Semantic Web and Linked Data world, there is a need to build tools that reuse existing visualization libraries showing the key information about RDF datasets. Many datasets are published using SPARQL Endpoints and are not "visually" accessible. Thus, understanding the underlying graphs and consuming them require lay users to have some knowledge in writing queries.
In this section, we propose a first step towards making available a semi-automatic way for the production of possible visualization of linked data sets of high-level categories grouping objects that are worth viewing and we associate them with some very well known vocabularies. Then, we describe the implementation of a Linked Data Visualization Wizard and its main components. This wizard can be used to easily visualize slices of datasets based on generic types detected.

5.2.2 Dataset Analysis

When developing an application, there are some *"important"* classes/categories, objects or datatypes that can be detected first to help to guide in the progress of creating a set of visualizations tied with those categories. We distinguish seven categories while acknowledging that this is not necessary an exhaustive list:

- § [**Geographic information**]: This category is for data modeled using `geo:SpatialThing`, `dbpedia-owl:Place`, `schema:Place` or `gml:_Feature` classes.
- § [**Temporal information**]: This category also includes dataset containing date, time (e.g: `xsd:dateTime`) and period or interval of time, using the OWL Time ontology.
- § [**Event information**]: This category is for any action of activity occurring at some place at some time.

- § [**Agent/Person information**]: This category is heavily influenced by the use of `foaf:Person` or `foaf:Agent`.
- § [**Organization information**]: This category is related to organizations or companies data, with the use of the `org` vocabulary[2] or the `foaf:Organization` class.
- § [**Statistics information**]: This category refers to statistical data generally modeled using the `data cube` vocabulary[3] or the SDMX model[4].
- § [**Knowledge Classification**]: This category refers to dataset describing schemas, classifications or taxonomies using the `SKOS` vocabulary.

5.3 Mapping Datatype, Views and Vocabularies

The On-line Library of Information Visualization Environments (OLIVE)[5] is a web site describing eight categories of information visualization environments differentiated by data type and collected by students, following a visualization course given at Maryland College Park, mostly inspired from the work of Ben Shneiderman [71]. Based on the classification provided by OLIVE, we propose a set of mappings between those categories (excluding the workspace dimension), views that can be applied to this category and a suitable list of vocabularies from the Linked Open Vocabularies catalogue [38][6] that correspond to those categories. Those vocabularies are easy to be found as there is a manual classification of vocabularies by the curators of the catalogue based on the content and scope of the terms and properties. According to the seven categories defined in Section 5.2.2, we have identified some of their corresponding one to one mapping with the set of vocabularies:

- **Geography** space, consisting of 21 vocabularies for features: `geo`, `gn`, `gf`, `om`, `geop`, `md`, `lgdo`, `loc`, `igeo`, `osadm`, `geod`, `ostop`, `place`, `geos`, `locn`, `coun`, `postcode`, `osr`, `geof`, `g50k` and `ad`.
- **Geometry** space, for vocabularies dealing with the geometries, mostly combined with the features, such as:
- **Time** space, consisting of 14 vocabularies, such as `cal`, `date`, `gts`, `interval`, `ncal`, `oh`, `te`, `thors`, `ti`, `time`, `tl`, `tm`, `tvc` and `tzont`.
- **Event** space, containing vocabularies such as `event`, `lode`, `music`, `sem`, `situ`, `sport`, `stories`, `theatre`, `tis` and `tisc`.

[2] `http://www.w3.org/TR/vocab-org/`
[3] `http://www.w3.org/TR/vocab-data-cube/`
[4] `http://sdmx.org/`
[5] `http://lte-projects.umd.edu/Olive/`
[6] `lov.okfn.org/dataset/lov/`

- **Government** space, with 9 vocabularies (`cgov`, `ctorg`, `elec`, `few`, `gc`, `gd`, `oan`, `odd`, `parl`) and the `org` vocabulary belonging to the W3C recommendation vocabularies at `http://lov.okfn.org/dataset/lov/lov#W3C`.

Metadata vocabularies, such as `rdfs`, `dcterms` or `dce` can be used in association with any of the visual element to give basic description of the resource of a giving dimension. For example, a popup information can be fired on a map view to display the relevant information of a geodata resource such as the label, the abstract or description. Another application can be to detect which visualization is best suited for geodata. Geodata belongs to a two-Dimension visual representation. Geodata is usually displayed using geographical-based visualizations (map, geo charts, etc.) and it is often modeled by vocabularies in the space named `Geometry` and `Geography`[7] vocabularies in RDF datasets. Hence those vocabularies can be combined to detect the presence or not of geographic information in a dataset, and thus yield to recommend a map view. Table 5.1 gives an overview of those mappings. For the tabular representation, it is the "default" visual representation of RDF data and can be used by any vocabulary without restriction.

Dimension	**Vocabulary Space**	**Visual element**
Temporal	Time space	TimeLine
one-Dimension	any	Tabular, text
two-Dimension	Geography space	Map view
	Geometry space	Maps view
three-Dimension	Event space	Map + TimeLine
Multi-Dimension	qb, sdmx-model, scovo	Charts, graphs
Tree	skos, Government space	Treemap, Org view
Network	any vocab.	Graph, network map

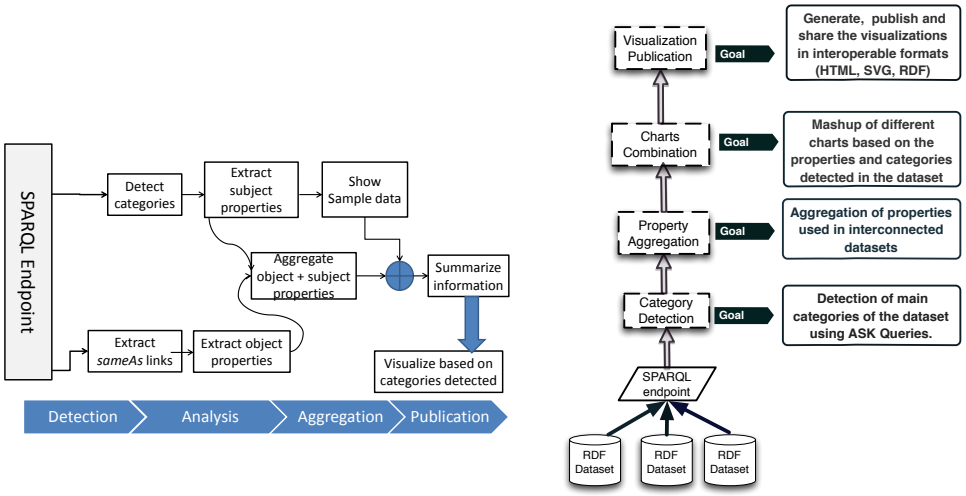
Table 5.1: A taxonomy of information visualization consuming Linked Datasets with associated views and suitable vocabulary space.

5.4 LDVizWiz: a Linked Data Visualization Wizard

We propose a workflow composed of four principal steps for a Linked Data Visualization Wizard, as depicted in Figure 5.1. Our requirement is to provide a tool that hide the complexity of SPARQL to lay users and at the same time, can be embedded in existing Linked Data infrastructure and workflow. First, we proposed to detect the presence of data belonging to one of the seven categories (Table 5.1) using generic SPARQL queries. More precisely, we perform `ASK` queries to test whether or not a particular query pattern has a solution. Second, we look at entities in a dataset that have `owl:sameAs` links with external objects and we retrieve the properties associated to those objects. We argue that the objects that are interlinked

[7] All the prefixes used for the vocabularies are the same used in LOV catalogue.

with other datasets are of primary importance in a visualization. We show the results of this mining process to the user (the categories that have been detected, the properties going with the categories and the external domain). Based on this information, the user can make a personalized "mashup" by aggregating the suitable visualization widgets. Some default visualizations are available according to the top categories detected. The last step is to publish the visualization and a metadata report in RDF/XML TURTLE or N3.



(a) The workflow of the different modules interacting in the Linked Data visualization wizard.

(b) High level functionalities of the Linked Data visualization wizard

Figure 5.1: Big picture and architecture of the Linked Data visualization wizard.

Let consider a graph $< G, c >$ to be $G = \{(s, p, o) | p \in URI, s \in URI, o \in (URI \cup LIT)\}$ where URI is the set of URIs, LIT is the set of literals, and c the context. We define $L = \{V_1, V_2, ..., V_n | V_i = P_i \cup T_i\}$ the list of vocabularies in LOV, with P_i and T_i respectively the properties and terms of a vocabulary V_i. Let also $D = \{D_1, D_2, .., D_m\}$ be the domains of vocabularies. We assume $\forall V \in L, \exists D_k \in \Phi(L, D)$. We define a generic function $\Sigma : (G, c) \mapsto B$ to detect categories in a dataset as follows: $\Sigma((G, c)) = \{B | (\exists (s, p, o) \in G : p \in V) \cup (\exists (s, rdf : type, o) \in G : o \in V)\}$ where $B = \{True, False\}$.
In the following sections, we describe each of the steps involved in the Linked Data Visualization Wizard in more details.

5.4.1 Category Detection

The goal of the category detection task is to use SPARQL queries to detect the presence of some high level categories in the dataset. We perform `ASK` queries as implementation of the Σ function using standard vocabularies as defined in the Table 5.1. We start with six categories, namely: geographic information, person,

organization, event, time and knowledge organization systems. We select popular vocabularies based on two existing catalogues: LOV [73] and prefix.cc[8].

```
ASK WHERE {
 {
  ?x a ?o.
  filter (?o= dbpedia-owl:Place ||
    ?o=gml:_Feature ||
    ?o=geo:SpatialFeature || ?o=gn:Feature ||
    ?o=admingeo:CivilAdministrativeArea ||
    ?o=spatial:Feature ||
    ?o=vcard:Location)
 }
 UNION {
  ?x ?p ?o. filter(?p=geo:lat || ?p=geo:long ||
  ?p=georss:point || ?p=geo:geometry ||
  geom:geometry)
 }
}
```

Listing 5.1: Generic query to detect geo data from a SPARQL endpoint

Listing 5.1 shows seven classes of different vocabularies are used, respectively for the namespaces `dbpedia-owl`, `geo`, `gn`, `admingeo`, `spatial` and `vcard`, with relevant classes to check the presence of geographic data.

```
  ASK WHERE {{?x a ?o. filter(?o=time:TemporalEntity ||
  ?o=time:Instant ||
    ?o=time:Interval || ?o=dbpedia-owl:TimePeriod ||
    ?o=time:DateTimeInterval || ?o=intervals:CalendarInterval)
  }
  UNION{ ?x ?p ?o. filter(?p=time:duration ||
  ?p=time:hasBeginning ||
  ?p=time:inDateTime || ?p=time:hasDateTimeDescription
  || ?p=time:hasEnd)}
```

Listing 5.2: Generic query to detect time data from a SPARQL endpoint, using `time`, `dbpedia-owl`, `intervals` vocabularies.

Listing 5.2 detects the presence of time information, while Listing 5.3, 5.4 and 5.5 detect persons, organizations and events respectively.

```
  ASK WHERE {?x a ?o. filter(?o = foaf:Person ||
   ?o=dbpedia-owl:Person ||
    ?o=vcard:Individual) }
```

Listing 5.3: Generic query to detect person categories from a SPARQL endpoint, using `foaf`, `dbpedia-owl`, `vcard` vocabularies.

```
PREFIX org:<http://www.w3.org/ns/org#>
  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
  ASK WHERE {?x a ?o . filter (?o=org:Organization ||
  ?o=org:OrganizationalUnit ||
  ?o=foaf:Organization ||
  ?o=dbpedia-owl:Organisation)}
```

[8] `http://prefix.cc`

Listing 5.4: Generic query to detect ORG data from a SPARQL endpoint.

```
1    ASK WHERE{?x a ?o. filter (?o= lode:Event || ?o=event:Event ||
2    ?o=dbpedia-owl:Event)}
```

Listing 5.5: Generic query to detect event data from a SPARQL endpoint, using `lode, event, dbpedia-owl` vocabularies.

For detecting data organized as taxonomy, `skos` vocabulary is used along with the most used classes and properties as showed in Listing 5.6.

```
1    ASK WHERE {{?x a ?o. filter(?o=skos:Concept ||
2     ?o=skos:ConceptScheme || ?o=skos:Collection )}
3       UNION{ ?x ?p ?o. filter(?p=skos:featureCode ||
4      ?p=skos:altLabel || ?p=skos:prefLabel  || ?p=skos:relatedMatch)}}
```

Listing 5.6: Generic query to detect SKOS data from a SPARQL endpoint, using `skos` vocabulary.

5.4.2 Property Aggregation

We take the benefits of the `owl:sameAs` links between entities to have access to the properties of the entities in the external namespaces different from the origin dataset. This module also aggregates the properties found in the dataset with the ones found in the interlinked sets. This is based on the assumption that during the linkage process, external datasets not only help in not breaking the *follow-your-nose* principle, but also add more information to be viewed in visualization applications. As shown in the code below, at this stage, we have collected and aggregated external properties gathered from the enrichment process of the workflow.

```
1-LET Namespace(?s) = S and LET Namespace(?t) =T
2-SELECT  owl:sameAs links
LET SEMTERM = list of ?s owl:sameAs ?t
WITH T != S
3-IN T, SELECT distinct properties used in dataset
4-AGGREGATE (3) with properties FROM S.
```

5.4.3 Visualization Generator

This module aims at recommending the appropriate visualizations based on the categories detected by the wizard. It might also help the user to make a report summarizing the result of the mining process, and then use different visualization libraries to view the data. This module can be viewed as a recommender system because it derives visualizations based on the categories. The input to build each visualization is the corresponding SELECT query of each ASK queries used to detect the categories. Moreover, some adjustment are made to avoid blank nodes and to

get the labels of the resources. The generator can be coupled with a mashup widget generator for some categories. For example, users could expect for event data, a combination of map view (where), a timeline (when) and facets based on the agents (who).

5.4.4 Visualization Publisher

The publisher module aims at exporting the combined visualizations, along with the report of all the process of mining the dataset, in a format easy to share, either as HTML, SVG or in the different RDF syntax flavor. For the latter, apart from using metadata information (creator, issued date, license), we model the categories we have detected using the `dcterms:subject` property of a `dcat:Dataset`, the queries used (using the `prov:wasDerivedFrom` property), the sample resources for each category (using the `void:exampleResource` property) and the visualization generated (using the `dvia` and `chart`[9] vocabularies).

5.5 Experiment and Implementation

In this section, we describe the experiments and report the evaluation on detecting categories on 444 endpoints. We then describe a prototype as a "`proof-of-concept`" of the proposal.

5.5.1 Experiment set up

We have evaluated our approach on the list of 444 endpoints referenced at `http://sparqles.okfn.org/` monitoring the availability, performance, interoperability and discoverability of SPARQL Endpoints registered in Datahub [74] . We have implemented a script in python to speed up the process and obtain the results. Every ASK query for the different category is implemented in a separate function requesting a JSON response.

5.5.2 Evaluation of the Category Detection

From the 444 endpoints used on the detection category module, 278 endpoints (62.61%) were able to give satisfactory (yes/no on one of the seven categories) results based on the queries. However, almost 37.38% of the endpoints were either down at the time of our experiments or the response header was in XML instead of JSON (as set up in the script). This result shows that our proposal with the current implementation (not covering all the vocabularies in LOV) make use of most popular vocabularies reused in the Linked Data.
This also implies a good coverage of the method that uses standard queries and yet can be extended. The full result of the detection module on the queried services is available at `http://cf.datawrapper.de/3FuiV/2/`, where for each column, the

[9] `http://data.lirmm.fr/ontologies/chart`

Category	number	Percentage
GEO DATA	97	21.84%
EVENT DATA	16	3.60%
TIME DATA	27	6.08%
SKOS DATA	2	0.45%
ORG DATA	48	10.81%
PERSON DATA	59	13.28%
STAT DATA	29	6.6%

Table 5.2: Classification of the endpoints according to the datatype detected with our SPARQL generic queries

value 0 stands for *no presence* and 1 for the *presence* of the categories. As provided in Table 5.2, 21.84% of geo data was detected, 13.288% of person data, 10.81% of org data and 3.6% of SKOS data.

Endpoint	event	geo	org	person	skos	time
dbpedia.org	0	1	1	1	0	0
de.dbpedia.org	0	1	1	1	0	0
el.dbpedia.org	1	1	1	1	0	0
fr.dbpedia.org	1	1	1	1	0	1
ja.dbpedia.org	1	1	1	1	0	0
live.dbpedia.org	1	1	1	1	0	1
nl.dbpedia.org	1	1	1	1	0	0
pt.dbepdia.org	1	1	1	1	0	0

Table 5.3: Categories detected in some *dbpedia* endpoint domains, where "1" is the presence and "0" the absence of the given type of category.

Table 5.3 summarizes some findings for 8 DBpedia chapters endpoints where it's easy to note the absence of SKOS data, and less presence of data modeled using `time` vocabulary. The Table also shows the differences in the standard vocabularies used to convert the Wikipedia data into RDF across different chapters.

5.5.3 Implementation

A first prototype, implemented with javascript and the Bootstrap framework[10], is available at `http://semantics.eurecom.fr/datalift/rdfViz/apps/`, as a proof of concept. We aim at providing a lightweight tool for lay users to quickly understand what the data is about and so that they get first visualizations based on categories detected in the datasets. We also reuse *sgvizler* [53] for generating charts according to the categories retrieved by the wizard. In the current implementation, the user

[10] `http://getbootstrap.com/`

can enter any SPARQL endpoint, and with a "click", the user can receive the list of categories detected together with sample resources. In the second step, the wizard retrieves the properties from the objects and subjects part of `owl:sameAs` links. The last step shows different tabs with the summary of the previous steps, the visualizations available for each categories, and a report both in human and machine readable formats. Figure 5.5.3 depicts a sample visualization generated by the wizard for geo data and statistics data.
The system can be used in any tool consuming Linked Data in which the complexity of SPARQL analysis and visualizations of RDF datasets is hidden to the lay users, with the benefits of showing that information encoded in triples is not only "beautiful", but also useful in the sense of traditional wizard-based tools.

Figure 5.2: Categories detected and visualization generated by the Linked Data visualization wizard in the case of EventMedia endpoint service.

5.6 GeoRDFviz: Map visualization of Geodata Endpoints

With the growing interest of publishing geolocation data according to linked data principles, many endpoints are provided without any visual interface to help navigating on top of the data. This lead to make it difficult for lay users to grab the essence of the data without learning some SPARQL queries. On the other hand, visual depiction of a location in a map makes it easier to identifier a resource. `GeoRDFviz` is a lightweight tool that help understanding the geodata resources in

any endpoint in a more attractive way for non-experts users. GeoRDFviz is built using generic queries in SPARQL and three visual actions: (i) zooming, (ii) filtering and (iii) describing of resources with geometry. GeoRDFviz prototype at `http://semantics.eurecom.fr/datalift/GeoRDFviz/`. The user first select the endpoint to visualize. GeoRDFviz picks up randomly a resource contained in the endpoint and direct the user to that position in a map. From this point, the user can zoom-in to find list of resources around that area of the map, and progressively can reach a more specific resource. By clicking on the resource, a pop-up menu shows more connections of the resource modeled as `SKOS` concepts. The application is a client application implemented using Backbone.js library[11].

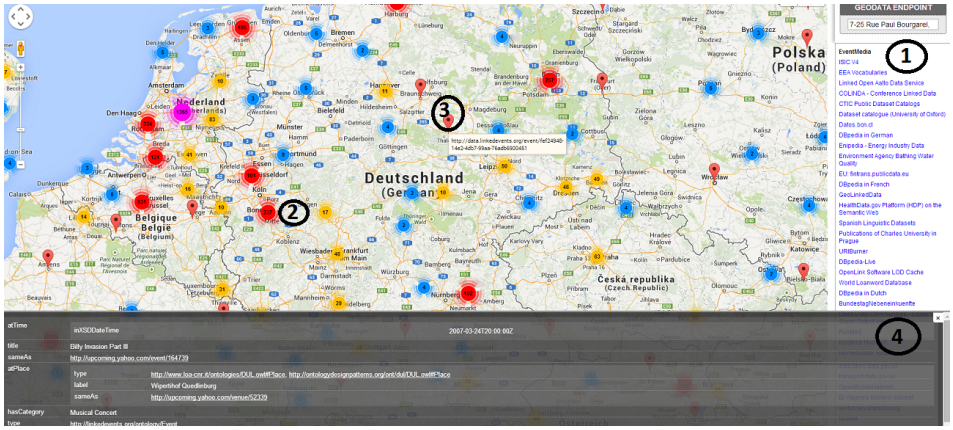


Figure 5.3: Screenshot of the user interface. The circles with numbers highlight the different elements : (1) list of endpoints, (2) number of resources available in the map area, (3) A zoom to a given element and (4) description of the selected resource.

5.7 A vocabulary for Describing VIsualization Applications

We have implemented a vocabulary, DVIA[12] aims at describing any applications developed to consume datasets in 4-5 stars, using visual tools to showcase the benefits of Linked Data. It reuses four existing vocabularies: Dublin Core terms at `http://purl.org/dc/terms/`, dataset catalogue (DCAT) at `http://www.w3.org/ns/dcat#`, Dublin Core Metadata Initiative at `http://purl.org/dc/dcmitype` and the Organization vocabulary at `http://www.w3.org/ns/org#`. It is composed of three main classes :

- **Application**: This class represents the application or the mashup devel-

[11] `http://backbonejs.org/`
[12] `http://bit.ly/Vb4L8k`

oped for demo-ing or consuming data in LD fashion. It is subclass of **dctype:Software**

- **Platform**: The platform where to host or use the application, could be on the Web (Firefox, Chrome, IE, etc..) or mobile (android, iOS, mobile) or even desktop

- **VisualTool**: Represents the tool or library used to build the application.

The diagram of the main classes and properties is depicted in Figure 5.4. The current version of the vocabulary in Turtle format can be found at `http://purl.org/ontology/dvia` . Listing 5.11 is a snapshot of the description of the application which won the Semantic Web Challenge[13] in 2012, the *EventMedia Live* application, described using DVIA vocabulary. It depicts apart from some metadata about the application (dct:title, dct:name, dct:issued, dct:creator and dct:license), the different visualization libraries integrated for building Eventedia Live (e.g.: Google API, Backbone, etc), as well as the operating systems where it is designed for, the different views/facets available in the application (map, charts, graphs, force-directed layout, ect) and the heteregeneous datasets used to implement it.

```
visuapp:eventMedia01
        a dvia:Application ;
        dct:title "EventMedia Live"@en;
        dvia:description "An application for reconciling Live events with
            media" ;
        dvia:url <http://eventmedia.eurecom.fr> ;
        dct:issued "2012-11-10"^^xsd:date ;
        dvia:businessValue "not commercial" ;
        dvia:keyword "events, media"^^xsd:string ;
        dct:license <http://www.opendatacommons.org/licenses/pddl/1.0/> ;
       dvia:platform [ a dvia:Platform ;
       dct:title: "Desktop" ;
       dvia:preferredNavigator "Google Chrome" ;
       dvia:alternativeNavigator "FireFox" ;
       dvia:system "Mac OS, Windows, Linux"^^string ] ;

      dvia:usesTool [ a dvia:visualTool; dct:title "Google visualization
          Tool" ;
   dct:description "Google visualization API" ;
   dvia:accessUrl <https://developers.google.com/chart/interactive/docs/
       reference> ;
   dvia:downloadUrl <http://www.google.com/uds/modules/gviz/gviz-api.js/> ]
        ;
     dvia:usesTool visuapp:visualTool02 ;
     dvia:consumes [ a dcat:Dataset; dct:title "BBC dump"] ;
     dvia:consumes [ a dcat:Dataset; dct:title "last.fm scrapped dataset"]
         ;
     dvia:consumes [ a dcat:Dataset; dct:title "upcoming scrapped dataset"]
         ;
     dvia:consumes [ a dcat:Dataset; dct:title "eventful scrapped dataset"]
         ;
     dvia:consumes [ a dcat:Dataset; dct:title "Flickr scrapped dataset"] ;
     dvia:consumes [ a dcat:Dataset; dct:title "Music Brainz "] ;
     dvia:consumes [ a dcat:Dataset; dct:title "Foursquare Json file"] ;
```

[13] `http://challenge.semanticweb.org/2012/winners.html`

```
28      dvia:consumes [ a dcat:Dataset; dct:title "DBpedia"] ;
29      dct:creator [foaf:mbox "khrouf@eurecom.fr"; foaf:name "Houda Khrouf"];
30      dct:creator [foaf:mbox "vuk@eurecom.fr"; foaf:name "Vuk Milicik"];
31      dct:creator [foaf:mbox "raphael.troncy@eurecom.fr"; foaf:name "Raphael
            Troncy"];
32      dvia:view "map, chart, graph, force-directed layout" ;
33          .
34       ...
```

Listing 5.7: Snapshot in Turtle of the description of Event Media Live Application

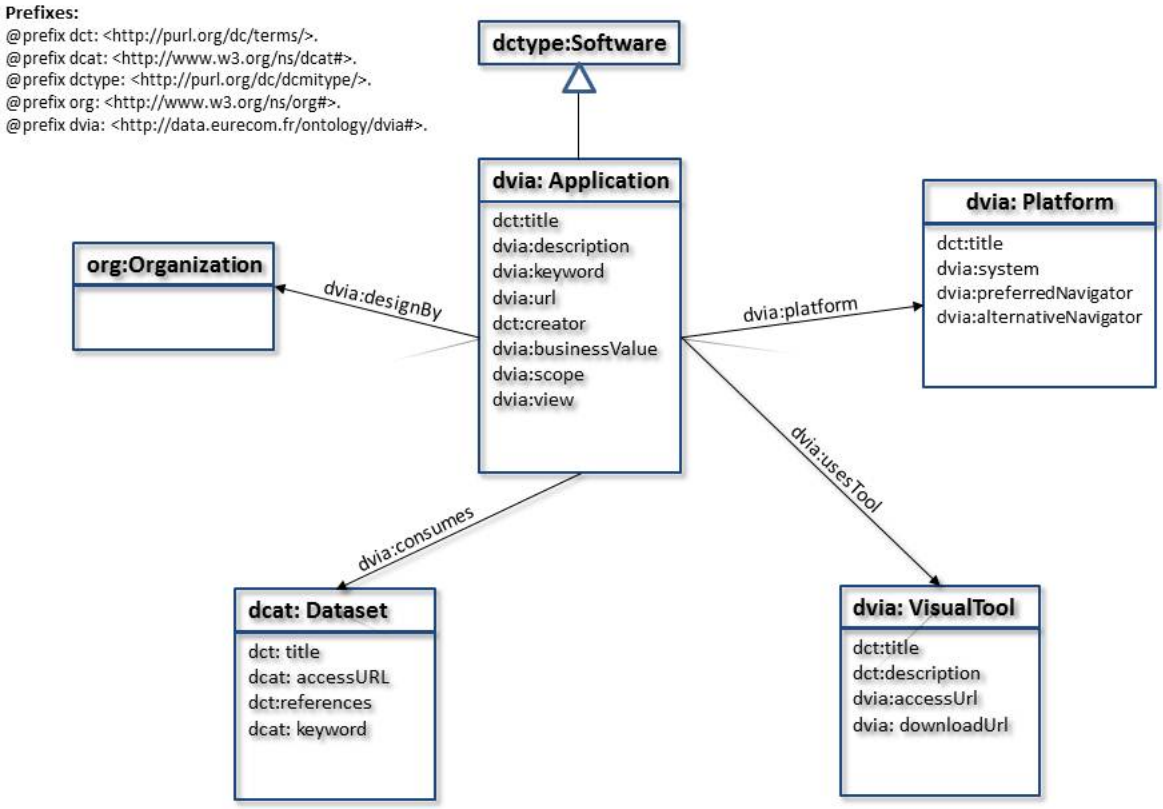


Figure 5.4: Conceptual Model of the DVIA vocabulary

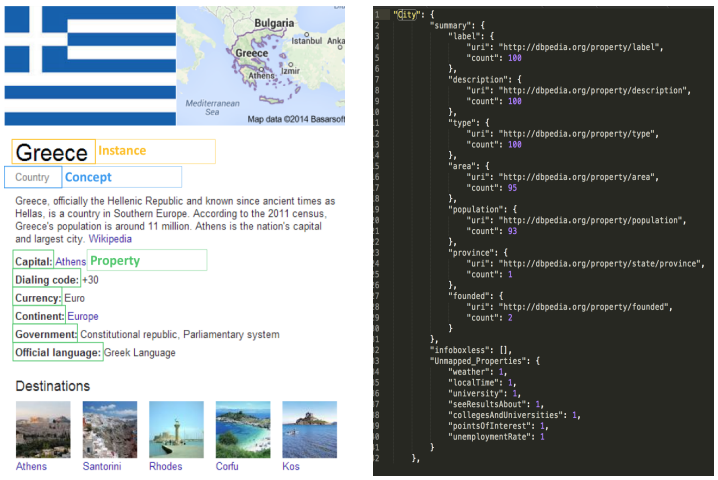
5.8 Important Properties for an Entity

In many knowledge bases, entities are described with numerous properties. However, not all properties have the same importance. Some properties are considered as keys for performing instance matching tasks while other properties are generally chosen for quickly providing a summary of the key facts attached to an entity. Our

motivation is to provide a method enabling to select what properties should be used when depicting the summary of an entity, for example in a multimedia question answering system such as QakisMedia[14] or in a second screen application providing more information about a particular TV program[15]. Our approach consists in: (i) reverse engineering the Google Knowledge Panel by extracting the properties that Google considers as sufficiently important to show (Section 5.8.1), and (ii) analyzing users' preferences by conducting a user survey and comparing the results (Section 5.8.2). We finally show how we can explicitly represent this knowledge of preferred properties to attach to an entity using the Fresnel vocabulary [75].

5.8.1 Reverse Engineering the Google KG Panel

Web scraping is a technique for extracting data from Web pages. We aim at capturing the properties depicted in the Google Knowledge Panel (GKP) that are injected in search result pages [76]. We have developed a Node.js application that queries all DBpedia concepts that have at least one instance which is `owl:sameAs` with a Freebase resource in order to increase the probability that the search engine result page (SERP) for this resource will contain a GKP. We assume in our experiments that the properties displayed for an entity are type and context dependent (country, time, query) which can affect the results. Moreover, we filter out generic concepts by excluding those who are direct subclasses of `owl:Thing` since they will trigger ambiguous queries. We obtained a list of 352 concepts[16].



(a) Google's Knowledge Panel for Greece

(b) Results of the crawling for the class City

Figure 5.5: Google Knowledge Graph Reverse Engineering Process.

[14] `http://qakis.org/`

[15] `http://www.linkedtv.eu/demos/linkednews/`

[16] See also the SPARQL query at `http://goo.gl/EYuGm1`

Algorithm 1 Google Knowledge Panel reverse engineering algorithm

```
1: INITIALIZE equivalentClasses(DBpedia, Freebase) AS vectorClasses
2: Upload vectorClasses for querying processing
3: Set n AS number-of-instances-to-query
4: for each conceptType ∈ vectorClasses do
5:    SELECT n instances
6:    listInstances ← SELECT-SPARQL(conceptType, n)
7:    for each instance ∈ listInstances do
8:       CALL http://www.google.com/search?q=instance
9:       if knowledgePanel exists then
10:         SCRAP GOOGLE KNOWLEDGE PANEL
11:      else
12:         CALL http://www.google.com/search?q=instance + conceptType
13:         SCRAP GOOGLE KNOWLEDGE PANEL
14:      end if
15:      gkpProperties ← GetData(DOM, EXIST(GKP))
16:   end for
17:   COMPUTE occurrences for each prop ∈ gkpProperties
18: end for
19: RETURN gkpProperties
```

For each of these concepts, we retrieve n instances (in our experiment, n was equal to 100 random instances). For each of these instances, we issue a search query to Google containing the instance label. Google does not serve the GKP for all user agents and we had to mimic a browser behavior by setting the $User - Agent$ to a particular browser. We use CSS selectors to check the existence of and to extract data from a GKP. An example of a query selector is $._om$ (all elements with class name $\_om$) which returns the property DOM element(s) for the concept described in the GKP. From our experiments, we found out that we do not always get a GKP in a SERP. If this happens, we try to disambiguate the instance by issuing a new query with the concept type attached. However, if no GKP was found again, we capture that for manual inspection later on. Listing 1 gives the high level algorithm for extracting the GKP. The full implementation can be found at `https://github.com/ahmadassaf/KBE`. We finally observe that this experiment is only valid for the English Google.com search results since GKP varies according to top level names.

Figure 5.5 shows the GKP for Greece, and the results of crawling and mapping for the class `City`. In this particular case, each of the properties of the GKP are directly mapped with the corresponding property in DBpedia and the frequency. Hence, a City is likely to be described in the GKP with at least the following properties: label, description, type, and population.

5.8.2 Evaluation

We conducted a user survey in order to compare what users think should be the important properties to display for a particular entity and what the GKP shows.

5.8.2.1 User survey.

We set up a survey[17] on February 25th, 2014 and for three weeks in order to collect the preferences of users in term of the properties they would like to be shown for a particular entity. We select only one representative entity for nine classes: `TennisPlayer`, `Museum`, `Politician`, `Company`, `Country`, `City`, `Film`, `SoccerClub` and `Book`. 152 participants have provided answers, 72% from academia, 20% coming from the industry and 8% having not declared their affiliation. 94% of the respondents have heard about the Semantic Web while 35% were not familiar with specific visualization tools. The detailed results[18] show the ranking of the top properties for each entity. We only keep the properties having received at least 10% votes for comparing with the properties depicted in a KGP. We observe that users do not seem to be interested in the `INSEE code` identifying a French city while they expect to see the `population` or the `points of interest` of this city.

5.8.2.2 Comparison with the Knowledge Graphs.

The results of the Google Knowledge Panel (GKP) extraction[19] clearly show a long tail distribution of the properties depicted by Google, with a top N properties (N being 4, 5 or 6 depending on the entity) counting for 98% of the properties shown for this type. We compare those properties with the ones revealed by the user study. Table 5.4 shows the agreement between the users and the choices made by Google in the GKP for the 9 classes. The highest agreement concerns the type `Museum` (66.97%) while the lowest one is for the `TennisPlayer` (20%) concept. We think properties for museums or books are more stable than for types such as person/agent which vary significantly. We acknowledge the fact that more than one instance should be tested in order to draw meaningful conclusion regarding what are the important properties for a type.

Classes	TennisPlayer	Museum	Politician	Company	Country	City	Film	SoccerClub	Book
Agr.	20%	66.97%	50%	40%	60%	60%	60%	50%	60%

Table 5.4: Agreement on properties between users and the Knowledge Graph Panel

With this set of 9 concepts, we are covering 301,189 DBpedia entities that have an existence in Freebase, and for each of them, we can now empirically define the most important properties when there is an agreement between one of the biggest knowledge base (Google) and users preferences.

5.8.2.3 Modeling the preferred properties with Fresnel.

Fresnel[20] is a presentation vocabulary for displaying RDF data. It specifies *what* information contained in an RDF graph should be presented with the core concept

[17] The survey is at `http://eSurv.org?u=entityviz`

[18] `https://github.com/ahmadassaf/KBE/blob/master/results/agreement-gkp-users.xls`

[19] `https://github.com/ahmadassaf/KBE/blob/master/results/survey.json`

[20] `http://www.w3.org/2005/04/fresnel-info/`

`fresnel:Lens`. We use the Fresnel and PROV-O ontologies[21] to explicitly represent what properties should be depicted when displaying an entity. This dataset can now be re-used as a configuration for any consuming application.

```
:tennisPlayerGKPDefaultLens rdf:type fresnel:Lens ;
  fresnel:purpose fresnel:defaultLens ;
  fresnel:classLensDomain dbpedia-owl:TennisPlayer ;
  fresnel:group :tennisPlayerGroup ;
  fresnel:showProperties (dbpedia-owl:abstract dbpedia-owl:birthDate
    dbpedia-owl:birthPlace dbpprop:height dbpprop:weight
    dbpprop:turnedpro dbpprop:siblings) ;
  prov:wasDerivedFrom
    <http://www.google.com/insidesearch/features/search/knowledge.html> .
```

Listing 5.8: Excerpt of a Fresnel lens in Turtle

5.9 Application consuming Event datasets: Confomaton

In this Section, we aim at creating a rich environment to enable users to navigate events as well as their various representative media such as photos, slides and tweets. A typical usage is to gather data about a scientific conference and investigate the added value of collecting scientific related media. A non trivial task in such application is to connect structured data with extremely noisy content, especially in the case of a major conference.

5.9.1 Background

We consider a scientific conference, the International Semantic Web Conference (ISWC 2011), which took place in Bonn, Germany in November 2011. Broadly speaking, considering all co-authors, people who have participated in the reviewing process, people who physically attended the conference or tried to follow it on social networks, we estimate that it attracted more than 1,500 participants. The conference organizers publish a lot of structured data regarding the conference including the list of accepted papers, their authors and institutions, the detailed program composed of sub-events with the exact timetable and the location (rooms) of the talks. This data is modeled using the `SWC` ontology[22], which is designed to describe academic events, and uses classes and properties from other ontologies such as `FOAF` (for people) and `SWRC` (BibTeX elements for the papers). The main conference of type `swc:ConferenceEvent` is related to a set of sub-events, namely (`WorkshopEvent`, `TutorialEvent`, `SessionEvent`, `TalkEvent`) via the property `swc:isSuperEventOf`. Table 5.5 shows some statistics about the data provided by the Dog Food server regarding the ISWC 2011 conference. We first notice that the data is incomplete. The conference has hosted 16 workshops in total, but the 75 papers are only associated to 8 of them while the 8 others did not have any papers according to the corpus. Furthermore, we find 133 papers that are not connected to any of the events via

[21] `http://www.w3.org/TR/prov-o/`

[22] `http://data.semanticweb.org/ns/swc/ontology`

the predicate `swc:hasRelatedDocument`. Finally, some useful information is also missing such as the keynote speakers and the *Semantic Web Death Match* (panel) event. This lack of knowledge is also a motivation for our work: can we collect and analyse social network activities in order to complete the factual description of this event?

Main Event	Sub-event	Number of events	Papers	Authors
Conference Event	Workshop Event	16	75	185
	Tutorial Event	7	7	20
	Session Event	1	66	202
	Talk Event	93	93	275
	-	-	133	385
Total (distinct)	4	117	292	735

Table 5.5: Metadata provided by the Dog Food Server for the ISWC 2011 conference.

We collected social network data in real time during the six days of the conference using the main tags advertised by the organizers (`#iswc2011`, `#cold2011`, `#derive2011`, etc.). Table 5.6 shows some statistics about the different media services used by the attendees along with the number of items from a number of distinct users. As expected, Twitter is by far the most used service: we have been able to collect 3,390 tweets from 519 different users. A significant proportion of tweets contains hyperlinks that we have further analysed. Hence, we extracted 384 different websites indexed by so-called URL shorteners (such as Bitly) found in 1,464 tweets (43% of tweets). These links represent a rich source of media, as they pointed to various Web resources categories such as blogs, slides, photos, publications and projects. For example, 25% of these links pointed to PDF documents that are generally one of the conference papers but could also be related papers relevant for the followers of the conference. We also analyse these links to extract the various media services used by Twitter.

Media Service	Items	Users
Twitter	3390 tweets	519
pic.twitter	12 photos	6
yfrog	10 photos	9
Twitpic	10 photos	6
Flickr	47 photos	6
Google+	30 posts	26
Slideshare	25 slides	20

Table 5.6: Media services used during ISWC 2011 conference

The name *Confomaton* is a word play on the French term *Photomaton* (English photo booth) and *conference*. Just like a Photomaton illustrates the scene inside of the photo booth, the *Confomaton* illustrates an event such as a conference enriched with social media. *Confomaton* is a Semantic Web application that produces and

consumes Linked Data and is composed of four main components (Figure 5.6): (i) an Event Collector that extracts events descriptions such as the ones available in the Semantic Web Dog Food corpus; (ii) a media collector that collects social media content and represents it in RDF using various vocabularies; (iii) a Reconciliation Module playing the role of associating social media with sub-events and external knowledge; (iv) a User Interface powered by an instance of the Linked Data API as a logical layer connecting all the data in the triple store with the front-end visualizations.

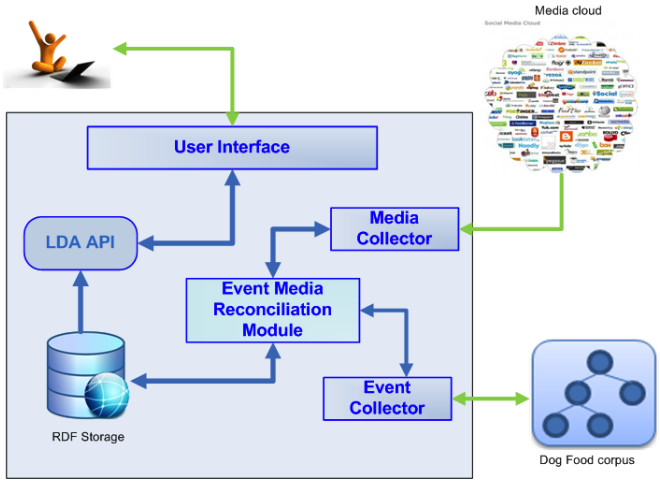


Figure 5.6: *Confomaton* general architecture.

5.9.2 Media Collector

In the context of *Confomaton*, we have developed the so-called *media collector* with the purpose of searching various social networks and media platforms for event-related media items such as photos, videos, and slides. We currently support 4 social networks (Google+, MySpace, Facebook, and Twitter) and 7 media platforms (Instagram, YouTube, Flickr, MobyPicture, img.ly, yfrog and Twitpic). Our approach being agnostic of media providers, we offer a common alignment schema for all of them:

- **Media URI**, the deep link to the media item, e.g. `http://farm7.staticflickr.com/6059/6290784192_567346ba6a_o.jpg`
- **Type**, the type of the media item, e.g. "photo"
- **Story URI**, the URI of the micropost or story where the media item appeared, e.g. `http://www.flickr.com/photos/96628098@N00/6290784192/`
- **Message**, the concrete micropost or description text in raw format, e.g. "Laura. `#iswc2011`, `#semanticweb`, `#bonn`, `#germany`"

- **Clean**, the concrete cleaned micropost or description text with some characters (e.g. hash signs) removed, e.g. "Laura. iswc2011, semanticweb, bonn, germany"

- **User**, the URI of the author of the micropost, e.g. `http://www.flickr.com/photos/96628098@N00/`

- **Published**, the timestamp of when the micropost was authored, or the media item was uploaded, e.g. `2011-10-27T12:24:41Z`

```
{
  "GooglePlus": [
    {
    "mediaurl": "http://software.ac.uk/sites/default/files/images/content/
        Bonn.jpg",
    "storyurl": "https://plus.google.com/107504842282779733854/posts/6
        ucw1Udb5NT",
    "message": {...}
    }],
    "Flickr": [
   {
      "mediaurl": "http://farm7.staticflickr.com/6226/6290782640
          _e8a1ffdcc2_o.jpg",
      "storyurl": "http://www.flickr.com/photos/96628098@N00/6290782640/",
      "message": {..}
    }]
}
```

Listing 5.9: Sample output of the media collector showing Google+ and Flickr results using #iswc2011 as query term

In order to retrieve data from media providers, we use the particular media provider's search Application Programming Interfaces (API) where they are available, and fall back to Web scraping the media provider's website if not. In some cases, we initially use the search API, but then have to fall back to Web scraping in order to get more details on the results, such as the **Media URI**, which is not exposed by all APIs. From all media providers, Twitter plays a special role, as it can serve as a host for other media providers. For example, it is very common for tweets to contain links to media items hosted on external media providers such as Twitpic. Other media providers treat media items as first class objects, i.e. have dedicated object keys in their API results for media items, which is not in all cases true for Twitter. We handle this by searching for a list of URIs of known media providers in combination with the actual search term. To illustrate this, when searching for media items for the search term "ISWC 2012" on Twitter, we would actually search for "iswc 2012 AND (twitpic.com OR flic.kr)" in the background, whereas on all other media providers, the search term "iswc 2012" is sufficient. The media collector can be tested at `http://webmasterapp.net/social/`.

5.9.3 Data Modeling of Confomaton

The Event Collector takes as input the Dog Food corpus described using the SWC ontology and converts all events into the LODE ontology[23], a minimal model that encapsulates the most useful properties for describing events. We use the `Room` ontology[24] for describing the various rooms contained in the conference centre. An explicit relationship between an event and its representative media (photo, slide, micropost, etc.) is realised through the `lode:illustrate` property. For describing those media, we re-use two popular vocabularies: the W3C Ontology for Media Resources[25] for photos and videos, and SIOC[26] for tweets, status, posts and slides. The example below shows how a tweet is represented in *Confomaton*.

```
<http://data.linkedevents.org/tweet/af557cef-5d5b-49c6-a4c3-bc9c41ce1555>
 a sioc:Post;
 dcterms:created "2011-10-23T13:34:03+00:00";
 sioc:content    "@smeh Good luck for your presentation at #ssn2011...";
 sioc:hasCreator <http://www.twitter.com/BadmotorF>;
 lode:illustrate <http://data.semanticweb.org/workshop/ssn/2011>;
 gc:hashtag      "#ssn2011";
 owl:sameAs      <http://twitter.com/BadmotorF/status/128071685235671040>.
```

Figure 5.7 depicts how all these vocabularies are used together. The ISWC 2011 conference is illustrated by a photo shared on Flickr, has for sub-event the EvoDyn 2011 workshop in which one of the tweets posted mentioned the recognised named entity Nathasha Noy who is also a general chair of the conference. All the data is available in the *Confomaton* graph in a public SPARQL endpoint available at `http://semantics.eurecom.fr/sparql`.

5.9.4 Event Media Reconciliation Module

The event media reconciliation module aims to align the incoming stream of social media with their appropriate events and to interlink some descriptions with general knowledge available in the LOD cloud[27] (e.g. people and institutions descriptions). Attaching social media to fine-grained events is a challenging problem. We tackle it by pre-processing the data with two successive filters in order to reduce the noise: one of them relies on keyword search applied to some fields such as title and tag, while the other one filters data based on temporal clues. The reconciliation is then ensured through a pre-configured mapping between a set of keywords and their associated events. This map enables us to associate media with the macro-events that people explicitly refer to in their posts or photos. For example, we connect all media items containing the tag `#iswc2011` with the general ISWC 2011 conference. However, this method is absolutely not convenient to associate media items with sub-events. For instance, in the ISWC 2011 conference, there are 99 sub-events of type `TalkEvent`, which could be the presentation of a paper, a keynote speech or any

[23] `http://linkedevents.org/ontology/`
[24] `http://vocab.deri.ie/rooms`
[25] `http://www.w3.org/TR/mediaont-10/`
[26] `http://rdfs.org/sioc/spec/`
[27] `http://lod-cloud.net/`

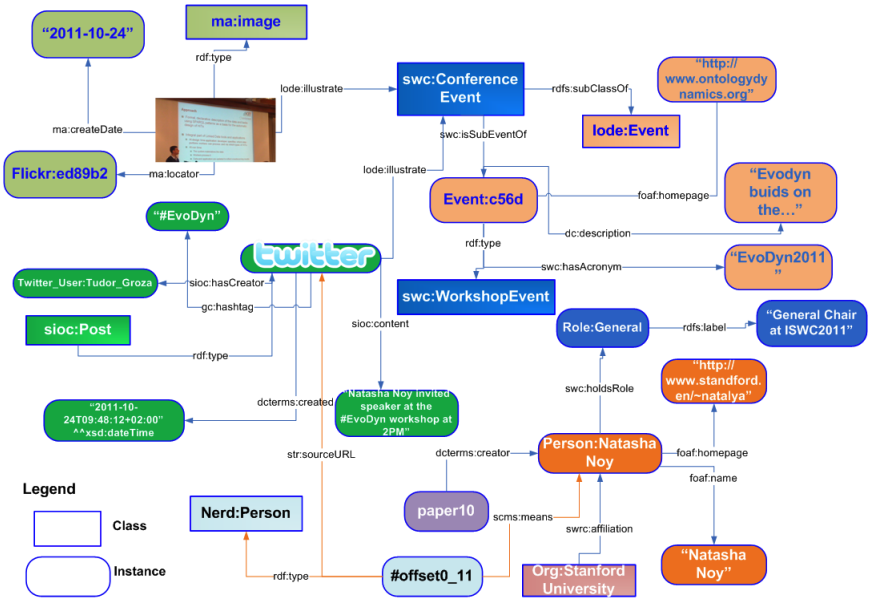


Figure 5.7: Example of data modeled in *Confomaton* re-using multiple vocabularies

other kind of talk. Social network users typically do not specify a particular tag for such events. We hence advocate the need for more advanced classifiers to associate media with sub-events. These classifiers can exploit a variety of parameters such as social network graphs and named entities extracted from media content.

5.9.5 Graphical User Interface

The Graphical User Interface (GUI) of *Confomaton* is built around four perspectives characterising an event: (i) *"Where does the event take place?"*, (ii) *"What is the event about?"*, (iii) *"When does the event take place?"*, and finally (iv) *"Who are the attendees of the event?"*. In addition, the user interface offers full text search for these four dimensions. The *Confomaton* user interface is powered by the Linked Data API[28], which provides a configurable way to access RDF data using simple RESTful URLs that are translated into queries to our SPARQL endpoint. More precisely, we use the Elda[29] implementation developed by Epimorphics. Elda comes with some pre-built samples and documentation, which allow to build specification to leverage the connection between the back-end (data in the triple store) and the front-end (visualizations for the user). The API layer helps to associate URIs with processing logic that extract data from the SPARQL endpoint using one or more SPARQL queries and then serialize the results using the format requested by the client. A URI is used to identify a single resource whose properties are to be retrieved or to identify a set of resources, either through structure of the URI, or through query parameters.

[28] `http://code.google.com/p/linked-data-api/wiki/Specification`
[29] `http://code.google.com/p/elda`

Listing 5.10 shows an example of the configuration file of the *Confomaton* API specifying the event, media and tweet viewers followed by the events and media properties access.

```
<#MyAPI> a api:API ;
  rdfs:label "Confomaton API"@en ;
  api:endpoint <#event>,<#media>,<#tweet>,<#agent>, <#venue>,<#user>,<#
      eventbyid>,<#mediabyid>,<#tweetbyid>,<#agentbyid>,<#venuebyid>,<#
      userbyid>;
  api:sparqlEndpoint <http://semantics.eurecom.fr/sparql> ;
# specification of the event viewer (all properties appear in the json
    file)
spec:eventViewer a api:Viewer ;
  api:property "title","description","space.lat","space.lon","time.
      datetime","inagent.label",...
<#eventbyid> a api:ItemEndpoint;
    api:uriTemplate "/event/{id}";
    api:itemTemplate "http://data.linkedevents.org/event/{id}";
    api:defaultViewer spec:eventViewer.
```

Listing 5.10: Example configuration file of the *Confomaton* API, specifying event properties access.

The system is available at `http://semantics.eurecom.fr/confomaton/iswc2011`. On the left side of the main view, the user can select the main conference event or one of the sub-events as provided by the Dog Food metadata corpus. In the centre, the default view is a map centred on where the event took place (e.g. Bonn, Germany) and the user is also encouraged to explore potential other types of events (concerts, exhibitions, sports, etc.) happening nearby, this data being provided by EventMedia [77]. The *What* tab is media-centred and allows to quickly see what illustrates a selected event (tweets, photos, slides). Zooming in an event triggers a popup window that contains the title and timetable of the event, the precise room location and a slideshow gallery of all the media items collected for this event. For the *When* tab, a timeline is provided in order to filter events according to a day time period. Finally, the *Who* tab aims at showing all the participants of the conference. This is intrinsically bound to a social component, aiming not only to present relevant information about participants (their affiliations, homepages, or roles at the conference), but also the relationships between participants themselves and with events.

5.10 Application consuming Statistics datasets

5.10.1 Scope of the Application

The `Perfect School` application is intended to provide useful information on schools in France using semantic technologies, with RDF-ized data enriched with other datasets in the wild. The application and the vocabulary have successfully passed the integrity checker[30] of an implementation for the candidate recommendation of Data Cube vocabulary [15], a recommendation from the W3C.

[30] `http://www.w3.org/2011/gld/wiki/Data_Cube_Implementations`

5.10.2 Legacy Datasets

In order to build the application, we had to look at some relevant datasets in data.gouv.fr. The ones selected for building the application are the following:

- The file at `http://www.data.gouv.fr/DataSet/564055` in CSV format, containing a list of 67,201 schools (name, status, type), with geolocation position in Lambert 93, for the academic year 2011-2012. The file contains the following attributes (in French):
  - code of the school (e.g. 0010002X),
  - official name of the school (e.g.: College Saint-Exupery),
  - principal name (e.g. COLLEGE),
  - patronymic name (e.g. Saint-Exupery),
  - status of the school (1 = open, 2 = to be closed and 3 = to be opened),
  - label of the type of school (e.g: 1= first degree, 3 = second degree).
- `http://www.data.gouv.fr/DataSet/572165`, in .CSV format giving results for professional schools, indicators), from one academic year (2011-2012). The file contains the following attributes:
  - name of school,
  - city code,
  - code of the school,
  - district where the school is located,
  - sector (PR= PRivate, PU= PUblic),
  - several observation measures with statistics on success rates, school versus academic/national rates, name of the academy it belongs to, as well as the department.
- `http://www.data.gouv.fr/DataSet/572162` (.CSV) containing statistics for 2296 public high schools and indicators. It complements the statistics from INSEE.

5.10.3 Ontology Modeling

We have reused some external ontologies for more interoperability:

- `aiiso`[31] for the type of school and codes of school.
- `geofla`[32] since the schools are considered as a topographic entities,

[31] `http://vocab.org/aiiso/schema`
[32] `http://data.ign.fr/def/geofla#`

- `geom`[33] for representing the different geometries (points with latitude and longitude) in a given coordinate reference systems with `ignf` ontology at `http://data.ign.fr/def/ignf#` .

- `skos:Concept` for describing the 30 types of nature of schools.

- `qb:DimensionProperty` and `qb:MeasureProperty` [15] for modeling the dimensions and different indicators available for a given school.

The resulting vocabulary is available at `http://purl.org/ontology/dvia/ecole`. We use the Datalift platform for transforming the different CSV files into RDF. The final data is available at `http://eventmedia.eurecom.fr/sparql` with the named graph `http://data.eurecom.fr/school`.

5.10.4 URI Policies

We use the following patterns for the URIs of the vocabularies or the resources in our namespace.

- URI for vocabulary: `http://data.eurecom.fr/ontologies/{SECTOR}`.
  e.g: `<http://data.eurecom.fr/ontologies/ecole>` for the ontology that we have developed.

- URI for resources: `http://data.eurecom.fr/id/{SECTOR}/{CLASS}`.
  e.g: `<http://data.eurecom.fr/id/school/>` for the schools URIs,
  `<http://data.eurecom.fr/id/school/slice/>` for `qb:Slice`s.

- URI for taxonomies: we use `SKOS` for modeling concepts and codes as the following:
  `http://data.eurecom.fr/codes/{SECTOR}/{CONCEPT-TYPE}/{CODE}`. e.g.:
  `<http://data.eurecom.fr/codes/ecole/natureUAI>` for the collection of natures of the schools ;
  `<http://data.eurecom.fr/codes/ecole/natureUAI/101>` for a particular concept with code *101* and label "'École maternelle".

Besides the aforementioned policy, each school in the user interface can be reached on the UI by using directly the following URI: `http://semantics.eurecom.fr/datalift/PerfectSchool/#school/{SCHOOL-CODE/}`, with {SCHOOL-CODE/} in lowercase.

Example: *The school "Albert Camus" in the city "Le Mans" with the code school* 0720800*D can we viewed in the application directly at* `http://semantics.eurecom.fr/datalift/PerfectSchool/#school/0720800d/`

[33] `http://data.ign.fr/def/geometrie#`

5.10.5 Sample School Data in RDF

```
1
2 school:0750676c   a aiiso:School, geofla:EntiteTopographique .
3 school:0750676c   a ecole:Etablissement .
4 school:0750676c rdfs:label  "LYCEE DORIAN (PROFESSIONNEL)"@fr ;
5 school:0750676c dcterms:title "Lycee polyvalent et lycee des metiers de la
      .."@fr ;
6   aiiso:code  "0750676C" ;
7   ecole:denominationPrincipale "LPO LYCEE DES METIERS"@fr ;
8   ecole:patronyme "DORIAN"@fr ;
9   ecole:ville "PARIS 11"@fr ;
10  ecole:codeCommune "75111" ;
11  ecole:secteur "PU" ;
12  ecole:academie  "PARIS"@fr ;
13  ecole:departement "PARIS"@fr ;
14  ecole:cycle "3"^^xsd:int ;
15  ecole:etat  "1"^^xsd:int ;
16  ecole:nature  bpenat:306 .
17
18 slice:0750676c  ecole:etablissement school:0750676c .
19 school:0750676c geom:geometrie  _:vb42480647 , _:vb42531960 .
20
21 _:vb42480647 a geom:Point;
22   geom:systCoord ignfr:wgs84g;
23   geom:coordX "48.85429801"^^xsd:double;
24   geom:coordY "2.39231163"^^xsd:double .
25
26 _:vb42531960 a  geom:Point;
27   geom:systCoord ignfr:ntflamb2e ;
28   geom:coordX "655410.1"^^xsd:double;
29   geom:coordY "6861755.9"^^xsd:double .
```

Listing 5.11: Snapshot in Turtle for the school ID=0750676C, also at `http://semantics.eurecom.fr/datalift/PerfectSchool/#school/0750676c/`

5.10.6 Interconnection

For the interconnection process, we didn't use the current module. Instead of that we have used the Silk [44] platform as it is re-packaged in the workflow of Datalift. We believe the scripts used for this task can be easily reused within the Datalift platform. Two datasets where used for finding `owl:sameAs` links:

1. DBpedia French chapter[34], as the scope of the application was limited to France. We have found only 7 match links with our schools datasets.

2. LinkedGeoData[35], as the underlying data used comes from the community project Open Street Map (OSM). Here, we got a total of 601 matching links in the category of `lgdo:BuildingSchool`[36].

[34] `http://fr.dbpedia.org/sparql`
[35] `http://linkedgeodata.org/sparql`
[36] `http://linkedgeodata.org/ontology/BuildingSchool`

5.10.7 User Interface

The target device for the application is Mobile phone, using principally two frameworks: Jquery mobile[37] and backbone javascript[38]. The application provides geolocation, search by city/district, graph charts for stats, table views of relevant results aggregated or group by some other aspects. `Perfect School Application` provides 3 main views:

1. **Search form:** The interface retrieves the location automatically (á la Google Maps API fashion) , and offers box choices based on the School type: First degree / second degree. When choosing first degree, the user can further select one of (primary school, elementary school or other). For the second degree, apart from looking for one of College, high school or other, the user can look for public or private schools. The search button launches the query behind the scene for retrieving the collection of data matching the user's criteria.

2. **Results of searching**: The search action returns a collection of schools plotted in a map. A cursor on the left side helps users zoom to get more details about schools retrieved in a given region, or street. When selecting a given school, the name is displayed and with the possibility to see the route from the barycenter of the result on the map.

3. **Description of the school:** It is divided in 3 different tabs: (a) General information (name, cycle, principal denomination, nature and patronym used) ; (b) Stats with all the different statistics in form of charts, graphs comparing the school with the others ; and (c) DBpedia-FR[39] information if available, obtained with the `owl:sameAs` links for enriching the original dataset with information such as founder, date of creation, web site, population, head of school etc.

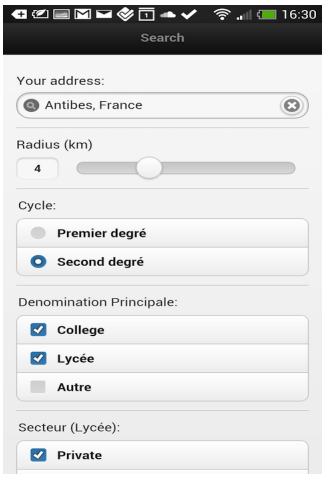
Figure 5.8 shows the three different views on a running example when using the application in a Mobile Phone.

5.11 Improving the discovery of applications contests in Open Data Events

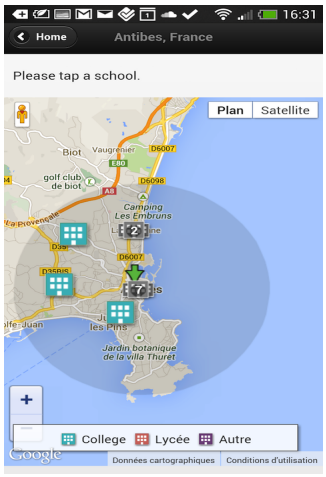
5.11.1 Background

One of the challenges in the Apps for Europe -project - and open data projects in general - is the discovery of existing applications using the open data. The discovery of existing applications and ideas is important in order to prevent people from reinventing the wheel, when they instead should put their effort in refining existing applications or developing completely new applications. This will hopefully

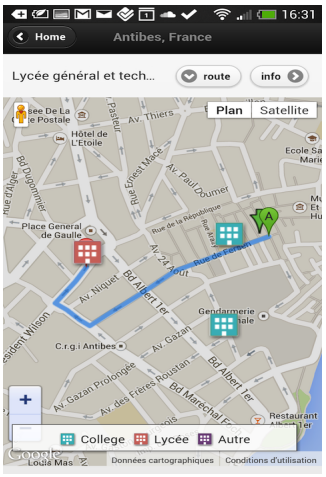
[37] http://jquerymobile.com/
[38] `http://backbonejs.org/`
[39] `http://fr.dbpedia.org`



(a) Search options



(b) Results displaying on a map



(c) Route to the selected school

(d) Details information with 3 tabs: General, statistics and DBpedia

Figure 5.8: Steps for searching high schools in Antibes in a radius of 4000m, France

lead to more diverse applications and furthermore also to higher quality applications, because the existing applications can be enhanced by other people and organizations [78].
In order to promote the reuse and discovery of open data applications, Apps for Europe has created an RDF vocabulary[40] that can be used for describing open data events and also the applications that have been built on top of the open data. Furthermore, Apps for Europe has also developed a Wordpress plugin[41] that open data event organizers can install on their web pages, to make it simpler to populate the RDF data for the events and applications.

5.11.2 Modeling events and applications in RDF

One of the key factors in improving the discovery of open data events is to present the information in a structured machine readable format, so that applications (such as crawlers) can easily consume the data. This follows the Semantic Web movement, where the goal is to convert the current, unstructured documents (readable by humans) into structured documents.
In order to present the event and application information in structured format, Apps for Europe has created an RDF vocabulary available a `https://github.com/mmlab/apps4eu-vocabulary/` for modeling the events and applications. The quality of this vocabulary was tested by manually modeling past events and applications and looking for potential problems and improvements in the ontology. The problems and improvements made to the ontology is presented below.

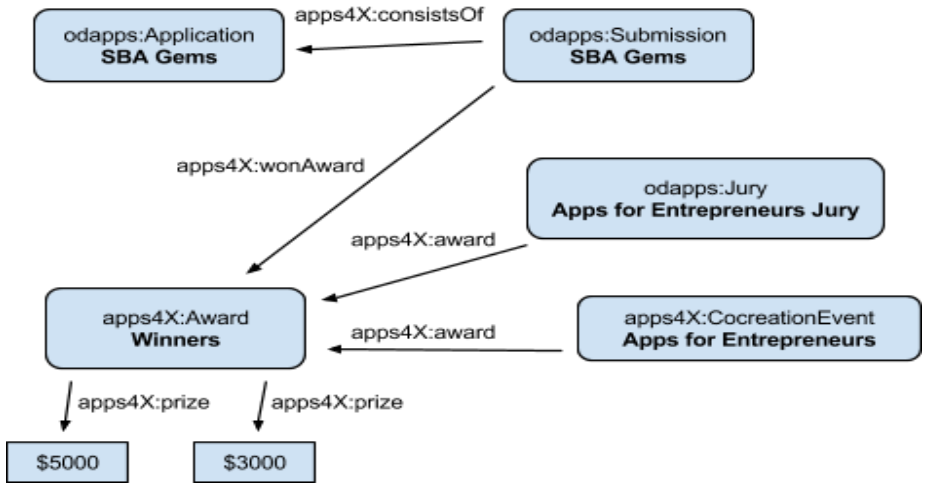


Figure 5.9: The RDF triples before changes. Here we state that the application SBA Gems has won an award in the event Apps for Entrepreneurs.

In 5.9, the model is complex, because it requires us to specify an extra class

[40] `https://github.com/mmlab/apps4eu-vocabulary`
[41] `https://github.com/mmlab/AppsForX`

(`:Submission`) in order to use the `:wonAward-property`, and yet we are unable to specify which prize the application won ($5000 or $3000). While in 5.10, the model is straightforward to specify that an application (SBA Gems) won a prize, because we don't need to use the Submission intermediate class. We are also able to specify which prize the application won (`:FirstPrize`). It should be notice also that now the jury is connected to a prize by specifying a jury-attribute for prizes (instead of having a prize-attribute for the jury i.e. direction of the arrow changed).

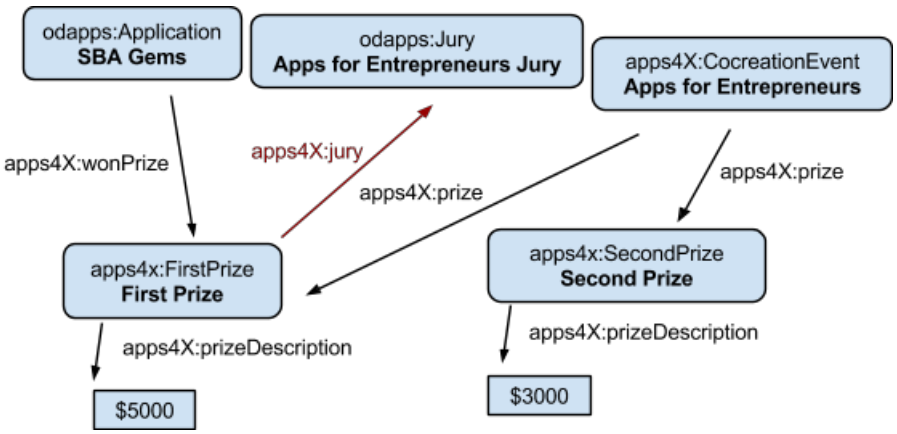


Figure 5.10: The RDF triples after changes.

5.11.3 Improving the model for specifying winners

Open data events often include competitions where the best applications are rewarded, and this information was modeled in the Apps for Europe RDF vocabulary as well. However, we found that the current model for this was unnecessarily complicated and therefore the model was simplified. Previously, winning apps required a separate "Submission"-class in order to state that they had won a competition. Furthermore, the vocabulary didn't allow to state which position (winner, second, third etc.) the winning application was awarded. These problems were fixed by introducing a new property "wonPrize" for applications, so that we can directly state that an application won a prize without needing to have an intermediary "Submission"-class, and by introducing the new classes "FirstPrize", "SecondPrize", and "ThirdPrize" that are all subclasses of the "Prize"-class. We concluded that it is satisfactory to be able to state positions one to three for the winning apps, and therefore we didn't implement a more complicated vocabulary that would have enabled to state an arbitrary position (by having a position attribute for the prize for example).
Also some other minor changes were suggested for the vocabulary:

- Use the word "Prize" instead of "Award" (i.e. rename in all places)
- Normalize the values for juryRate and usersRate attributes (because otherwise

they can't be compared to each other)

- alternatively reuse the review vocabulary from vocab.org[42] or schema.org[43]
- Introduce properties `connectedApp` (for Event) and `connectedEvent` (for Application), so that applications and events can be linked together.

5.11.4 Universal JavaScript plugin for RDF population

The inspiration for the JavaScript based solution came from services like `https://muut.com/`, which offers embeddable commenting and discussion forums and it can be installed on any web page independent of the CMS. Since almost all people have JavaScript support enabled in their browsers[44] it enables us to implement a solution that can be installed on any web page and also works on almost any browser. The most striking difference between the CMS based plugin and the JavaScript plugin is the user base, since the JavaScript solution works on any web page while the CMS plugin obviously only can be installed on a dedicated websites. Also the technical approach differs significantly, because in the CMS based approach the server will do all the work for producing the HTML and RDFa markup for the events, while in the JavaScript approach it is the client browser that is responsible for this task.

5.11.4.1 Technical description

The JavaScript plugin consists of three distinct parts (Cf. Figure 5.11) that are technically independent of each other, but they can still be deployed on the same server if needed:

RESTful API for creating, editing and removing events and applications from the database. Both interfaces listed below (2 and 3) communicates and manipulates the data using the RESTful API

Admin interface for event organizers. Event organizers manage their events and applications through this interface, which is provided as a "software as a service" (SaaS[45]).

Embeddable script that will display the event and application information both in human readable form and in computer readable form (RDFa format) on the event organizers web page. The script will fetch the event and application information using the RESTful API and then manipulate the document object model (DOM) after page load and inject the event and application description into the DOM.

[42] `http://vocab.org/review/terms.html`

[43] `http://schema.org/Review`

[44] According to a study by Yahoo only about 1% have JavaScript disabled in their browsers. Source: `https://developer.yahoo.com/blogs/ydnfourblog/many-users-javascript-disabled-14121.html` [Referenced 2014-06-02]

[45] `http://en.wikipedia.org/wiki/Software_as_a_service`

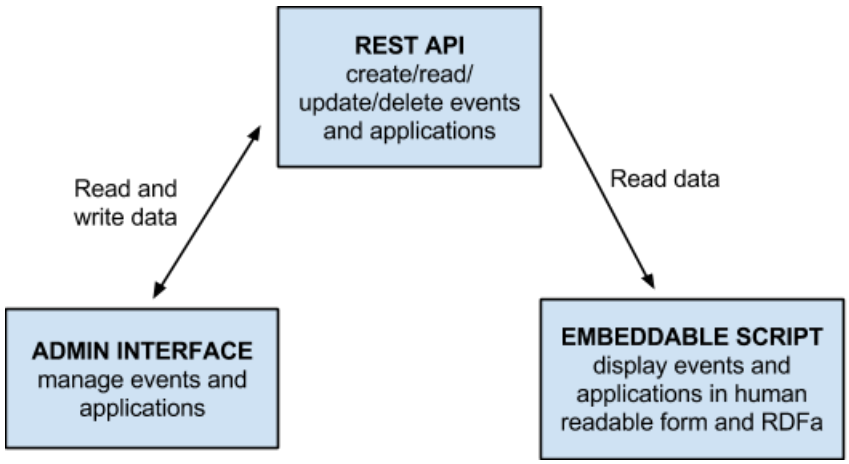


Figure 5.11: Universal JavaScript components

RESTful API: The API was implemented in Node.js[46] , which is a platform for running applications written in JavaScript on server side. It is built on Google Chrome's JavaScript runtime and provides very good performance thanks to it's asynchronous data manipulation model . Node.js has become very popular recently and there is a huge number of third party extensions and frameworks written for it. Furthermore, since the programming language is JavaScript, many libraries and frameworks written for browsers will also function in Node.js. For more details on how to configure the plugin, the readers are encouraged to read the Appendix A.
The data is stored in MongoDB, which is the leading NoSQL database[47]. In addition to this Mongoose[48] object data mapper was used in order to simplify the data manipulation and to give structure to the database. Finally, node-restful was used to transform the Mongoose schema into a working REST API. Node-restful internally relies on Express , which is the most popular web application framework for Node.js.

Admin Interface: The admin interface provides an intuitive and easy-to-use interface for event organizers to manager their events and the related applications. It is provided as software as a service and will be hosted and managed by Apps for Europe. Event organizers will register to the service and after this they have access to all the features of the admin interface. Two screenshots of the service is presented in Figure 5.12 .
The admin interface is implemented in HTML, CSS and JavaScript. Bootstrap[49] was used as a CSS framework. It simplifies the development of responsive websites and also provides clean aesthetics for the website. AngularJS[50] was used as a JavaScript

[46] http://nodejs.org/
[47] http://www.mongodb.com/leading-nosql-database
[48] http://mongoosejs.com/index.html
[49] http://getbootstrap.com/
[50] https://angularjs.org/

framework.

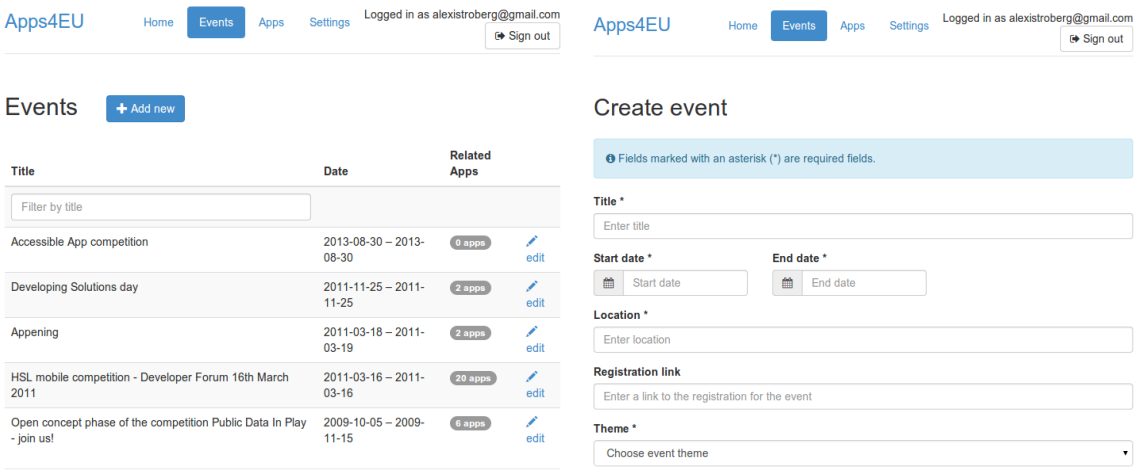


Figure 5.12: Screenshots of the admin interface. On the left is the event listing page and on the right is the form for creating an event.

Embeddable script The last component of the puzzle is the embeddable script, that is responsible of displaying the event and application information in human readable form on the event organizer's website. The event organizer will get the embed code from the admin interface, which he will then insert into the HTML code of his website. The embed code is just a reference to an external JavaScript file that will be loaded from the Apps for Europe server and then executed in the browser. A detailed description of how the script works is given below:

---

1. The event organizer will embed the code on his own web page.
2. On page load, the <script>-tag is parsed by the browser and the actual script is loaded from the Apps for Europe -server.
3. The script is executed.
(a). The script will fetch the event or application information using an AJAX-request.
(b). After the server has responded with the event/application information, the information is displayed on the page in human readable form by modifying the DOM of the page. Furthermore, the same information is presented also in computer readable format by embedding RDFa in the DOM.

5.11.5 Creating the Knowledge-base for past events

The final task was to create the knowledge base for past events by using the Apps for Europe RDF vocabulary and generating data to feed the endpoint at `http:`

//apps4europe.eurecom.fr/sparql . The list of events was extracted from the Apps for Europe Google spreadsheet[51].
The goal was to populate the events and applications, so that they would follow and fulfill the Apps for Europe vocabulary as well as possible. However, the format and information provided on the event web pages varied, and therefore we were usually only able to populate the following information for most of the events:

- Title
- Dates (start and end date)
- Description (free text)
- Prizes (if the event included an application contest)
- Jury members (if the event included an application contest)
- Location (usually only country was known)

Similarly, the information provided for applications varied even more, and sometimes we were able only to populate the title, description and the homepage of the application. Because of these problems, the data populated into the triple store doesn't include all the information we would have hoped to have.
The original idea was to automate the triple store data population as far as possible using content scraping , but because of the heterogeneity of the web pages (in terms of structure and data) a semi-automated approach was used. Especially the problems listed below prevented a fully automatic approach from being implemented:

- because of the heterogeneity in the structure of the web pages, it is for example extremely hard to "automatically" know which part of the page contains the event location or application homepage link
- many links to the event pages were broken, and thus manual work was required in order to relocate the actual page

In order to be as efficient as possible in the data population the semi-automatic approach was used. Here, the key idea was to populate event pages having only a few application entries (less than 10) manually using the JavaScript plugin, and for the rest of the event pages web scraping was used to populate the application entries (the scraping script had to be configured for each event website separately). More details of the script can be found in Appendix A.
The goal was to populate all the past events (112 in total). However, many of the websites for the past events had already disappeared or didn't include enough information for data population. Furthermore, the data population process turned out to take much more time than anticipated, and therefore we managed to populate only

[51] https://docs.google.com/spreadsheet/ccc?key=0AiXRLGASq8I0dDlfZURkWGpSODBJaWotQUp3eGNwNGc&usp=sharing

28 events and in total 889 application entries. The average number of application entries per event was 34 and the median 22.
Visualizations and applications could be built on top of the knowledge base, but because of the small size of the current knowledge base, it is difficult to extract quantitative data from the knowledge base. Some ideas for visualizations are presented below:

- Map visualization of where past events have been organized
- Most popular categories/themes for applications
- Gallery of application screenshots (visual inspiration for developers).

The dataset is available below in different formats as dump for download in RDF/-Turtle and MongoDB:

- RDF in turtle format: `https://www.dropbox.com/s/3075qsblxzau2fk/rdfInTurtleFormat.tar.gz`
- MongoDB dump: `https://www.dropbox.com/s/m2sr4na12v3yk07/apps4europe.tar.gz`

5.11.6 Discussion

The embeddable script is non-optimal in the sense that the event or application information is loaded only after the actual page (where the script is embedded) has loaded. This causes some additional delay before the page is fully rendered, but the problem can be alleviated by showing loading indicators.
CSS styling for the events and applications is provided using Bootstrap. All CSS rules have been made specific to the container div-element of the plugin, in order to prevent the CSS from conflicting with the CSS of the page. In the future, another solution called shadow DOM could be used to prevent conflicts between different components of the page, but at the moment the browser support for shadow DOM[52] is not satisfactory. Since the event and application information is directly embedded on the page using DOM, the event organizer can add his own CSS styling to the plugin by overriding the provided CSS rules.
The RDF information about events and applications is populated only if event organizers actually use it on their websites, and therefore it is crucial to ensure that the plugin is appealing in the eyes of the event organizers. It would be especially useful to study the usability of the plugin and also discuss with the event organizers how well the Apps for Europe ontology fits with their data model. In addition to this, also the security of the JavaScript-plugin should be improved, although the information entered to the plugin is not very critical from a security perspective.

[52] `http://caniuse.com/shadowdom`

5.12 Summary

In this chapter we have presented an approach for creating visualizations on top of Linked Data based on Semantic Web technologies. We first defined seven categories of objects worth viewing in a dataset, and we propose to associate them with commonly used and domain vocabularies. We then present a description of the main components of a Linked Data Visualization Wizard. We describe a lightweight implementation in JavaScript as a *proof-of-concept* of our proposal, with the benefits to be usable on-line or being extensible. We advocate that such a tool can be easily integrated in any workflow/framework for publishing and linking data on the Web, such as Datalift or the GeoKnow Stack. Besides, we have performed experiments on GKP to look for important properties in entities, and evaluated against users' preferences. Then we presented two applications in the domain of statistics and events, consuming different datasets in RDF on real-world scenario. We discussed on how to improve applications developed for contests, by proposing a vocabulary and a tool for populating the model by using a universal plugin. Some past events have been already semi-automatically curated using both the vocabulary and the plugin.

Part III

Contribution to Standards

Chapter 6

Best Practices for Publishing Linked Data

"Cool URIs don't change"
Tim Berners-Lee

6.1 Introduction

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 have provided some guidance to aid in the access and re-use of open government data. Based on the ability of Linked Data to provide a simple mechanism for combining data from multiple sources across the Web, it also addresses many objectives of open government transparency initiatives through the use international Web standards for the publication, dissemination and reuse of structured data [3].
To publish data on the web, stakeholders have to follow different tasks sometimes includes in different life cycle models. As described in [3], different proposals of life cycles all share common activities, summarized in the need to specify, model and publish data in standard open Web formats. Below are four different life cycles and their specificities:

- Hyland et al.[79] provide a six-step "cookbook" to model, create, publish, and announce government linked data. They highlight the role of the World Wide Web Consortium (W3C) which is currently driving specifications and best practices for the publication of governmental data. Hyland et al. lifecycle consists of the following activities: (1) Identify, (2) Model, (3) Name, (4) Describe, (5) Convert, (6) Publish, and (7) Maintain.

- According to Hausenblas et al.[80] existing data management approaches assume control over schema, data and data generation, which is not the case in the Web because it is open, de-centralized environment. Based on their experience in Linked Data publishing and consumption over the past years, they identify involved parties and fundamental phases, which provide for a multitude of so called Linked Data life cycles that consist of the following steps: (1)

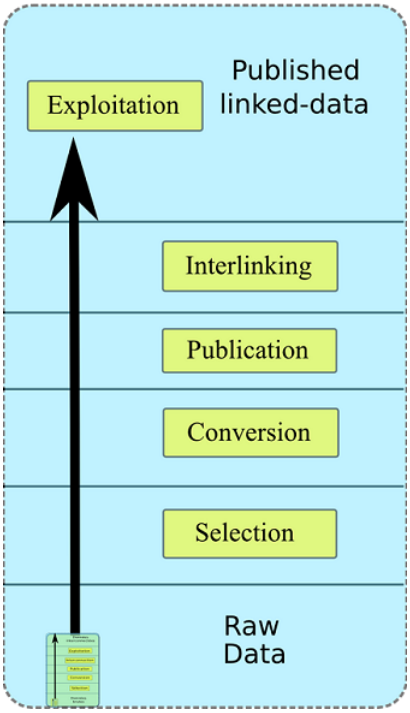


Figure 6.1: Datalift life cycle for publishing linked data

data awareness, (2) modeling, (3) publishing, (4) discovery, (5) integration, and (6) use cases.

- Villazón-Terrazas et al. propose in [81] a first step to formalize their experience gained in the development of government Linked Data, into a preliminary set of methodological guidelines for generating, publishing and exploiting Linked Government Data. Their life cycle consists of the following activities: (1) Specify, (2) Model, (3) Generate, (4) Publish, and (5) Exploit.

- Datalift vision [37], the process is divided into 3 principal phases: modeling the data , publishing and exploitation. Figure 6.1 depicts the different steps, closed to the architecture of the platform:

  1. Modeling the data consists of: (1) supporting the data selection, (2) identifying the relevant vocabulary, (3) defining a schema pattern for the URIs and (4) converting between formats
  2. Publishing the dataset consists of: (1) interconnecting with external datasets, (2) attaching provenance metadata information, (3) managing access right to the dataset and (4) storing the data in a triple store
  3. Exploitation, such as visualizing the dataset, re-publishing and versioning.

Although the process of lifting raw data to interlinked data in Datalift seems to

be linear, it can also be cyclic with a maintenance task in the life cycle. Table 6.1 summarizes the ten best practices to be taken into account to publish data as Linked Data on the Web.
In this chapter, we mainly describe our contributions on concrete implementations regarding how to use standard vocabularies (Section 6.3, Section 6.4 and Section 6.5), how to model and convert the data (Section 6.6, Step #3 & Step #7) and practical consideration on how to specify appropriate license on vocabularies and datasets (Section 6.7, cf. Step #4).

6.2 Catalog of Vocabularies

In [82], a vocabulary is a collection of "terms" with more or less complex semantics used to model a domain. Vocabularies can range from simple such as the widely used RDF Schema, FOAF and Dublin Core Metadata Element Set to complex vocabularies with thousands of terms, such as those used in healthcare to describe symptoms, diseases and treatments. Vocabularies play a very important role in Linked Data, specifically to help with data integration. It's use can also overlaps with Ontology in the context of Linked Data. Regarding catalog of vocabularies, while we refer the reader to [4] for a systematic survey of ontology libraries, we give our own classification of ontology repositories (Table 6.2). In particular, we distinguish six categories of catalogs:

- *Catalogs of generic vocabularies/schemas* similar to the LOV catalog, but without any relations among the vocabularies. Example of catalogs falling in this category are vocab.org[1], ontologi.es[2], JoinUp Semantic Assets or the Open Metadata Registry.

- *Catalogs of ontologies for a specific domain* such as biomedicine with the BioPortal[3], geospatial ontologies with SOCoP+OOR[4], Marine Metadata Interoperability and the SWEET ontologies[5].

- *Catalogs of ontologies from a project* such as the famous DAML repository of ontologies[6].

- *Catalogs of ontology Design Patterns (ODP)* focused on reusable patterns in ontology engineering.

- *Catalogs of editors' ontologies* used to test some features of a tool and to keep track of the ontologies built by a tool, such as Web Protégé or TONES.

[1] `http://vocab.org/`
[2] `http://ontologi.es/`
[3] `http://bioportal.bioontology.org/`
[4] `http://socop.oor.net/`
[5] `http://sweet.jpl.nasa.gov/2.1/`
[6] `http://daml.org/ontologies/`

- *Catalogs of ontologies maintained by a single organization* which often uses a platform such as Neologism[7] for publishing vocabularies.
- *Vocabularies crawled by Semantic Web search engines* containing snapshots at the time of the crawsuch as Watson[8], Sindice[9], Falcon-s[10] or Swoogle. For example, the NanJing Vocabulary Repository (NJVR)- a dump of Falcon-s ontologies, reported as of June, 17th 2,996 vocabularies crawled from 261 pay-level domains.

[7] http://neologism.deri.ie
[8] http://watson.kmi.open.ac.uk/
[9] http://www.sindice.com
[10] http://ws.nju.edu.cn/falcons/

Step	Name	Description
STEP #1	PREPARE STAKEHOLDERS	Prepare stakeholders by explaining the process of creating and maintaining Linked Open Data
STEP #2	SELECT A DATASET	Select a dataset that provides benefit to others for reuse.
STEP #3	MODEL THE DATA	Modeling Linked Data involves representing data objects and how they are related in an application-independent way
STEP #4	SPECIFY AN APPROPRIATE LICENSE	Specify an appropriate open data license with a clear statement about the origin, ownership and terms related to the use of the published data.
STEP #5	GOOD URIs FOR LINKED DATA	Consider a good URI naming strategy and implementation plan, based on HTTP URIs. Consideration for naming objects, multilingual support, data change over time and persistence strategy are the building blocks for useful Linked Data.
STEP #6	USE STANDARD VOCABULARIES	Describe objects with previously defined vocabularies whenever possible. Extend standard vocabularies where necessary, and create vocabularies (only when required) that follow best practices whenever possible.
STEP #7	CONVERT DATA	Convert data to a Linked Data representation, typically done by script or other automated processes.
STEP #8	PROVIDE MACHINE ACCESS TO DATA	Provide various ways for search engines and other automated processes to access data using standard Web mechanisms.
STEP #9	ANNOUNCE NEW DATA SETS	Remember to announce new data sets on an authoritative domain. Importantly, remember that as a Linked Open Data publisher, an implicit social contract is in effect.
STEP #10	RECOGNIZE THE SOCIAL CONTRACT	Recognize your responsibility in maintaining data once it is published. Ensure that the dataset(s) remain available where your organization says it will be and is maintained over time.

Table 6.1: Summary of the best practices to publish linked data on the Web adapted from [3]

Catalog name	Number of vocabularies	Search Feature	Category	Vocabulary maintenance
vocab.org	19	No	Catalog of generic vocabularies	N/A
ontologi.es	39	No	-//-	N/A
Joinup Semantic Assets	112	Yes	-//-	Yes
Open Metadata Registry	308	Yes	–//–	Yes
BioPortal	355	Yes	Catalog of Domain vocabularies	Yes
SOCoP + OOR	40	Yes	–//–	Yes
Marine Metadata Interoperability	55	Yes	–//–	Yes
SWEEET 2.2	200	No	–//–	N/A
DAML	282	No	–//–	No
ODPs	101	No	Catalog of ODPs	Yes
vocab.derie.ie	68	No	Catalog of Organizations	Yes
data.lirmm.fr ontologies	15	No	–//-	Yes
TONES	219	No	Catalog of editors' vocabularies	N/A
Web Protégé	69	No	–//–	Yes

Table 6.2: Catalogs of vocabularies with respectively the number of the ontologies, the presence of a search feature, the catalog category and whether it is maintained or not

We observe that the existing catalogs of vocabularies in the literature have some limitations compared with LOV. In terms of coverage, the number of vocabularies indexed by LOV is constantly growing and it is the only catalog, to the best of our knowledge, that provide all types of search criteria (metadata search, within/across ontologies search), both an API and a SPARQL endpoint access and that can be as well classified as an "Application platform" apart from being at the same time an ontology directory and an ontology registry. According to the categories of ontology libraries defined in [4], LOV falls under the category of "curated ontology directory" and an "application platform" because the ontologies are curated manually with statistics automatically generated, and because it exposes its data via an API. Furthermore, LOV provides an answer to some of the issues mentioned in the survey reported in [4], such as "where has an ontology been used before?" or "is this ontology compatible with mine?". In particular, LOV provides vocabulary usage statistics of the LOD Cloud datasets and it exposes vocabularies dependency using the Vocabulary-of-A-Friend (VOAF) ontology.
vocab.cc[11] is a service which is similar to prefix.cc since it enables to look up and search for Linked Data vocabularies while providing more specific information about the usage of a particular class or property in the Billion Triple Challenge Dataset (BTCD). It also provides the ranking of those properties or classes. The authors mentioned that "common prefixes are resolved with data from prefix.cc". Although they don't give further details, this service is somehow related to prefix.cc. TripleChecker[12] is a web service based on prefix.cc which aims at finding typos and common errors in RDF data. It parses a given URI/URL and the output is divided in two sections: the namespaces and the term section. The former matches against prefix.cc to determine whether they are "common prefixes" and the latter provides the term definition.

6.3 Linked Open Vocabulary (LOV) and Vocabularies

The Linked Open Vocabularies (LOV) initiative aims to bring more insights about published vocabularies in order to foster their reuse. Compared to other projects, LOV benefits from a community:

- to assess the quality (including documentation, metadata) and the reuse potential of a vocabulary before it is indexed. LOV contains currently 350+ reusable and well-documented vocabularies;

- to augment vocabularies with explicit information not originally defined in the RDF vocabulary. For example, only 55% of vocabularies have explicit metadata of at least one creator, contributor or editor. In LOV, we augmented this information leading to more than 85% of vocabularies with this information;

[11] `http://vocab.cc`
[12] `https://github.com/cgutteridge/TripleChecker`

- to automatically extract the implicit relations between vocabularies using the Vocabulary Of Friend[13] (VOAF) ontology. These relations can be used as a new metric for ranking terms based on their popularity at the schema level;
- to consider vocabulary semantic in the result ranking: a literal value matched for the `rdfs:label` property has a higher score than for the `dcterms:comment` property.

The way vocabularies are considered in LOV is similar to the way datasets are considered in the LOD cloud [83]. Hence, while the Vocabulary of Interlinked Datasets (VoiD) is used to describe relationships between datasets and their vocabularies [84], VOAF is used to describe the mutual relationships between vocabularies. VOAF itself reuses over popular vocabularies such as Dublin Core Terms (dcterms), Vocabulary Of Interlinked Datasets (VoiD), Vocabulary for ANNotating vocabulary (vann) and the BIBliographic Ontology (bibo). The vocabulary also introduces new classes such as `voaf:Vocabulary` and `voaf:VocabularySpace`.
The LOV-Bot is the tool that automatically keeps up-to-date the relationships and the metadata about the vocabularies indexed in LOV, using the following steps:

- LOV-Bot daily checks for vocabularies update (any difference in the vocabulary formal description fetched using content negotiation);
- LOV-Bot uses SPARQL constructs to detect relationships and metadata and creates explicit metadata descriptions in the LOV dataset;
- LOV-Bot annotations are then listed in a back-office administration dashboard in order to be reviewed. This manual part enables LOV curators to interact with vocabularies authors and the wider community to raise issues and make remarks or suggestions.

The LOV dataset is synchronized with the information presented in the web site. The latter allows a human user to browse LOV information. The Linked Open Vocabularies initiative does not only monitor the current state of the ecosystem. It also aims at storing and giving access to vocabularies history. To achieve this goal, the LOV database contains every different version of a vocabulary over the time since its first issue. For each version, a user can access the file and a log of modifications since the previous version.

6.3.1 Linked Open Vocabularies

Started in March 2011, in the framework of the DataLift research project [37] hosted by the Open Knowledge Foundation, the Linked Open Vocabularies (LOV) initiative is now standing as an innovative observatory of the vocabularies ecosystem. It gathers and makes visible indicators not yet harvested before, such as interconnection between vocabularies, versioning history and maintenance policy, past and current

[13] `http://lov.okfn.org/vocab/voaf/`

referent (individual or organization) if any. The number of vocabularies indexed by LOV is constantly growing (390 as of January 2014) thanks to a community effort and it is the only catalog, to the best of our knowledge, that provide all types of search criteria (metadata search, within/across ontologies search), both an API and a SPARQL endpoint access. According to the categories of ontology libraries defined in [4], LOV falls under the category of *"curated ontology directory"* and *"application platform"*.
The development of LOV has highlighted a number of interesting research issues such as *"where to find the best domain vocabulary to reuse?"*, and *"is it possible to create a curated catalogue of vocabularies that are links?"* Below we illustrate some of the LOV features useful for ontology search and reuse activities:

Domain filtering. Each vocabulary is inserted into LOV according to its domain and/or scope. This information is guided by the scope of the vocabulary, such as City, Science, Library, Metadata, Media, etc. This feature helps in disambiguating the results of the querying service and to classify vocabularies.

Content aware Search. If the searched term matches a `rdfs:label` it will have a higher score than if it matches `dcterms:comment`.

Links between vocabularies. One of the key feature of LOV design is the explicit links between vocabularies,

Scope of LOV. The intended use is to promote and facilitate the reuse of vocabularies in the linked data ecosystem.

Vocabulary Curation. The collection of the vocabularies is maintained by curators in charge of validating and inserting vocabularies in the LOV ecosystem, by taking care of the versions of the vocabulary and giving some reviews. The vocabulary is then automatically enriched with more information about the datasets using it, and relations to other vocabularies.

LOV focuses only on vocabularies (subpart of semantic documents of the web) submitted by any user, reviewed and validated by curators. In addition, LOV keeps track of different versions of the vocabularies in the server that can be retrieved for comparing the differences between along the time evolution. In contrast, Swoogle is designed to automatically discover Semantic Web Documents (SWDs), indexes their metadata and answers queries about it. Thus, the result of a search query retrieved any semantic document. For example, a query of the term *person* gives $16,438$ results while in LOV, only the term appears in 134 vocabularies. Watson works similarly to Swoogle, crawling and indexing semantic document at a small scale, explicitly distinguishing for each document (resource), concepts, properties and individuals if available. While in Swoogle the ranking score is displayed, Watson shows the language of the resource and the size. Falcons is a keyword-based search system for concepts and objects on the Semantic Web, and is equipped with entity summarization for browsing. It is notable that Falcons limits the search only

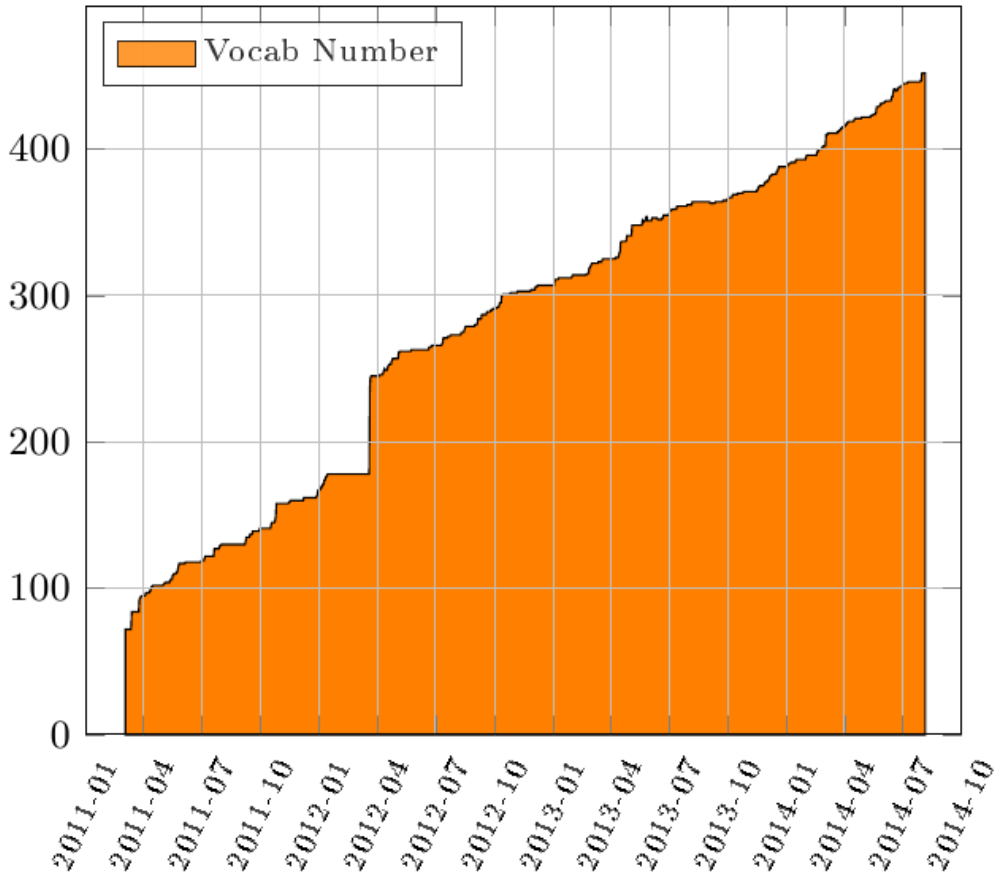


Fig. 3. Evolution of the number of vocabularies in LOV.

Figure 6.2: Graph evolution of vocabularies inserted into LOV since from 2011 to 2014.

to ontologies and a recommendation feature is provided according to users' preferences. However, it does not provide any relationships between the related ontologies, nor any domain classification of the vocabularies. Table 6.3 lists some key features of LOV with respect to Swoogle, Watson and Falcons.

6.3.2 LOV vs Neon Methodology

The NeOn Methodology is a scenario-based methodology that supports the collaborative aspects of ontology development and reuse, as well as the dynamic evolution of ontology networks in distributed environments. The key assets of the NeOn Methodology are [2]:

- A set of nine scenarios for building ontologies and ontology networks, emphasizing the reuse of ontological and non-ontological resources, the re-engineering and merging, and taking into account collaboration and dynamism.

Feature	Swoogle	Watson	Falcons	LOV
Browsing ontologies	Yes	Yes	Yes	Yes
Scope	SWDs	SWDs	Concepts	ontologies
Metrics	Ranking	Ranking	Ranking	LOD popularity
Domain filtering	No	No	No	Yes
Comments and review	No	Yes	No	Only by curators
Ranking	Doc. based	Doc. based	Doc. based	Metric-based
Web service access	Yes	Yes	Yes	Yes
SPARQL endpoint	No	No	No	Yes
Read/Write	Read	Read & Write	Read	Read
Ontology directory	No	No	No	Yes
Application platform	No	No	No	Yes
Storage	Cache	-	-	Dump & endpoint
Interaction with Contributors	No	-	No	Yes

Table 6.3: Comparison of LOV, with respect to Swoogle, Watson and Falcons; based on part of the framework defined in [4].

- The NeOn Glossary of Processes and Activities, which identifies and defines the processes and activities carried out when ontology networks are collaboratively built by teams.
- Methodological guidelines for different processes and activities of the ontology network development process, such as the reuse and reengineering of ontological and non-ontological resources, the ontology requirements specification, the ontology localization, the scheduling, etc.

LOV is a catalog and API that can fits well within the Neon methodology for building vocabularies and ontologies. Based on the Neon Methodology's glossary of activities for building ontologies, LOV is relevant in four activities:

Ontology Search. Main LOV's feature is the search of vocabulary terms. These vocabularies are categorized within LOV according to the domain they address. In this way, LOV contributes to ontology search by means of (a) keyword search and (b) domain browsing.

Ontology Assessment. LOV provides a score for each term retrieved by a keyword search. This score can be used during the assessment stage and includes a unique term statistical feature[14] which provides for each term registered in LOV the following information: (a) "LOV distribution" that represents the number of vocabularies in LOV that refer to a particular element; (b) "LOV popularity" that shows the number of other vocabulary elements that refers

[14] `http://lov.okfn.org/dataset/lov/stats/`

elem1	alignment	elem2
http://xmlns.com/foaf/0.1/Agent	http://www.w3.org/2002/07/owl#equivalentClass	http://purl.org/dc/terms/Agent
http://xmlns.com/foaf/0.1/maker	http://www.w3.org/2002/07/owl#equivalentProperty	http://purl.org/dc/terms/creator

Figure 6.3: Equivalent classes and properties between foaf and dcterms

to a particular one; and (c) "LOD distribution" that refers to the number of datasets in LOD which use a particular vocabulary; and (d) "LOD popularity" that refers to the number of vocabulary element occurrences in the LOD.

Ontology Mapping. As explained in section **??**, vocabularies rely on each other in seven different ways. In LOV these relationships are explicitly stated using VOAF vocabulary. This data could be useful to find alignments between ontologies, for example one user might be interested in finding equivalent classes for a given class or all the equivalent classes among two ontologies. The following example shows the retrieved data when asking for all the equivalent classes and properties between the vocabularies foaf and dcterms by means of the related VOAF query[15]:

```
SELECT DISTINCT ?elem1 ?alignment ?elem2 {
     {?elem1 <http://www.w3.org/2002/07/owl#equivalentClass> ?elem2}
     UNION {?elem1 <http://www.w3.org/2002/07/owl#equivalentProperty> ?elem2}
     UNION {?elem2 <http://www.w3.org/2002/07/owl#equivalentClass> ?elem1}
     UNION {?elem2 <http://www.w3.org/2002/07/owl#equivalentProperty> ?elem1}
     FILTER(!isBlank(?elem2))
     FILTER(!isBlank(?elem1))
     ?elem1 ?alignment ?elem2.
     ?elem1 rdfs:isDefinedBy <http://xmlns.com/foaf/0.1/>.
     ?elem2 rdfs:isDefinedBy <http://purl.org/dc/terms/>.
} ORDER BY ?alignment
```

Figure 6.3 shows the alignments between foaf and dcterms vocabularies by mean of `owl:equivalentClass` and `owl:equivalentProperty`.

Ontology Localization. Labels in different languages are stored in the LOV endpoint for the ontology terms that provide such information. This annotations could be used when translating terms into different languages. This information could be extracted by querying the SPARQL endpoint as in the following example[16] where all the labels defined for the terms that have at least one `rdfs:label` containing strictly "person":

```
SELECT DISTINCT ?label2 ?element{
     ?element rdfs:label ?label1 .
     ?element rdfs:label ?label2 .
```

[15] `http://goo.gl/sTIGQ6`. Prefixes are omitted for readability purpose. You can find the correct namespace for a prefix in LOV.

[16] `http://goo.gl/JJCJ01`

"Person"	http://xmlns.com/foaf/0.1/Person
"Persona"@es	http://xmlns.com/foaf/0.1/Person
"Personne"@fr	http://xmlns.com/foaf/0.1/Person
"Person"@en	http://xmlns.com/foaf/0.1/Person

Figure 6.4: Translations example for foaf:Person

```
      FILTER (?label1 != ?label2 ).
      FILTER(REGEX(STR(?label1), "person", "i")).
} ORDER BY ?element
```

An excerpt of the query result is shown in figure 6.4. From that result, "Persona"@es and "Personne"@fr could be used as translations for the English term "Person" in Spanish and French respectively.

Figure 6.5 shows the activities LOV can support within the overall Neon methodologies activity workflow.

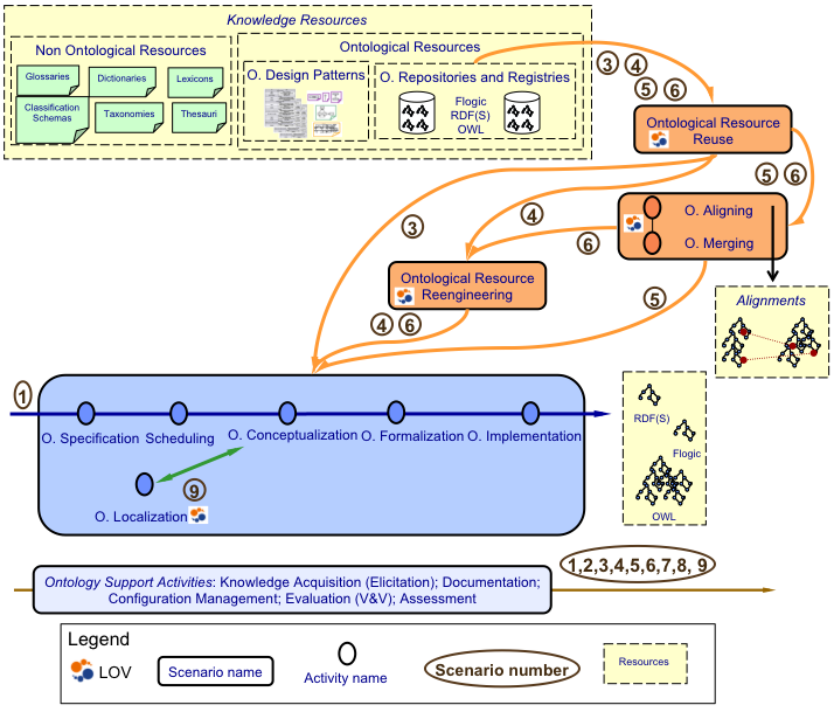


Figure 6.5: Meeting points between LOV and the NeOn methodology, derived from [2].

6.4 Prefixes harmonization

RDF vocabularies bring their meaning to linked data by defining classes and properties, and their formal semantics. Relying on W3C standards RDFS or OWL, those vocabularies are a fundamental layer in the architecture of the Semantic Web. Without the explicit semantics declared in vocabularies, linked data, even using RDF, would be just linked pieces of information where links have no meaning. Interoperability between data and datasets rely heavily on shared vocabularies, but given the distributed nature of the Web, vocabularies are published by independent parties and there is no centralized coordination of this publication, nor should it be. Various independent services have been developed in order to discover vocabularies and provide information about them, and the community of data publishers and vocabulary managers have all interest in complementarity and coordination between such services. In this section, we focus on a specific aspect of vocabularies: their identification by namespaces and associated prefixes.

In the original XML syntax of RDF, prefixes are simply local shortcuts associated with XML namespaces using `xmlns` declarations. The usage of prefixes has been further extended to other syntaxes of RDF such as N3 and Turtle. Although a prefix to namespace association is syntactically limited to the local context of the file in which it is declared, common prefixes such as `rdf:`, `rdfs:`, `owl:`, `skos:`, `foaf:` and many more have become de facto standards. For example, RDFa 1.1 has a default profile made of 11 well-used vocabularies based on their general usage on the Semantic Web according to the crawl of Yahoo! and Sindice as of March 2013[17]. Similarly, the YASGUI SPARQL editor has a list of built-in prefix-namespace associations to ease the construction of SPARQL queries. However, this list of "standard" prefixes is open-ended. Interfaces such as SPARQL endpoints (e.g. Virtuoso) use a list of built-in prefixes declaration for more and more namespaces but the choice of entries in this list is all but transparent. Hence, the reason of a given namespace being or not in this list could be interpreted in many ways, a potential source of technical and social conflicts. Therefore, the notion has been slowly spreading, at least implicitly, that common prefixes could and indeed should have a global use, implying some kind of governance and good practices. More and more vocabularies explicitly recommend the prefix that should be used for their namespace, generally using a common if not written good practice to avoid frontal clashes by recommending a prefix not already used. But there is no global policy except implicit rules of fair use to avoid potential conflicts resulting from polysemy (different namespaces using or recommending the same prefix) or synonymy (different prefixes used for the same namespace).

A vocabulary publisher needs to have access to some services capable of monitoring the existing prefixes usage in order to stick to those rules. Moreover, we focus on two services providing such information on prefixes usage namely prefix.cc[18] and LOV (Linked Open Vocabularies) [85]. Both services provide associations between

[17] `http://www.w3.org/2010/02/rdfa/profile/data/`

[18] Service: `http://prefix.cc/`; Code: `https://github.com/cygri/prefix.cc`

prefixes and namespaces but following a different logic. The prefix.cc service allows anybody to suggest a prefix to namespace association. It supports polysemy and synonymy, and has a very loose control on its crowd-sourced information. What it provides is more a measure of popularity of prefixes and namespaces than a way to put order in them. LOV has a much more strict policy forbidding polysemy and synonymy, enforced by a dedicated back-office database infrastructure, ensuring that each vocabulary in the LOV database is uniquely identified by a prefix, this unique identification allowing the usage of prefixes in various LOV publication URIs. This requirement leads sometimes to a situation where LOV uses prefixes different from the ones recommended by the vocabulary publishers.

6.4.1 Aligning LOV with Prefix.cc

In this section, we present how we perform the alignment between the two services LOV and prefix.cc. Figure 6.6 shows the evolution of the number of prefixes registered in these two services between April 2009 and July 2013. Our main goals are to align Qnames (prefix) to a unique URI in LOV and to make sure that all the vocabularies in LOV are actually inserted in prefix.cc.
We propose to perform SPARQL queries over all the files of prefix.cc at `http://prefix.cc/popular/all.file.vann` in the FROM clause and compare them to the content of the LOV SPARQL endpoint[19] via a SERVICE[20] call. The SERVICE keyword defined in the SPARQL 1.1 Query Language instructs a federated query processor to invoke a portion of a SPARQL query against a remote SPARQL endpoint [86]. Results are returned to the federated query processor and are combined with results from the rest of the query. To be more generic and standards-compliant, the queries could be run with the Jena ARQ command-line tool to produce a CSV or a JSON serialization that could be easily consumed either by the prefix.cc backend via phpMyAdmin or by the LOV backend.

6.4.2 First Task: prefixes in LOV not present in Prefix.cc

First, we compute $< LOV > INTERSECTS < PREFIX.CC >$ and $< LOV > MINUS \{< LOV > INTERSECTS < PREFIX.CC >\}$. The following SPARQL query finds namespace URIs in LOV that do not exist in prefix.cc along with their LOV prefix.

```
PREFIX vann: <http://purl.org/vocab/vann/>
SELECT ?prefix ?lovURI
FROM <http://prefix.cc/popular/all.file.vann> {
  SERVICE <http://lov.okfn.org/endpoint/lov> {
    SELECT ?prefix ?lovURI {
      [] vann:preferredNamespacePrefix ?prefix;
        vann:preferredNamespaceUri ?lovURI;
```

[19] `http://lov.okfn.org/endpoint/lov`
[20] `http://www.w3.org/2009/sparql/docs/fed/service`

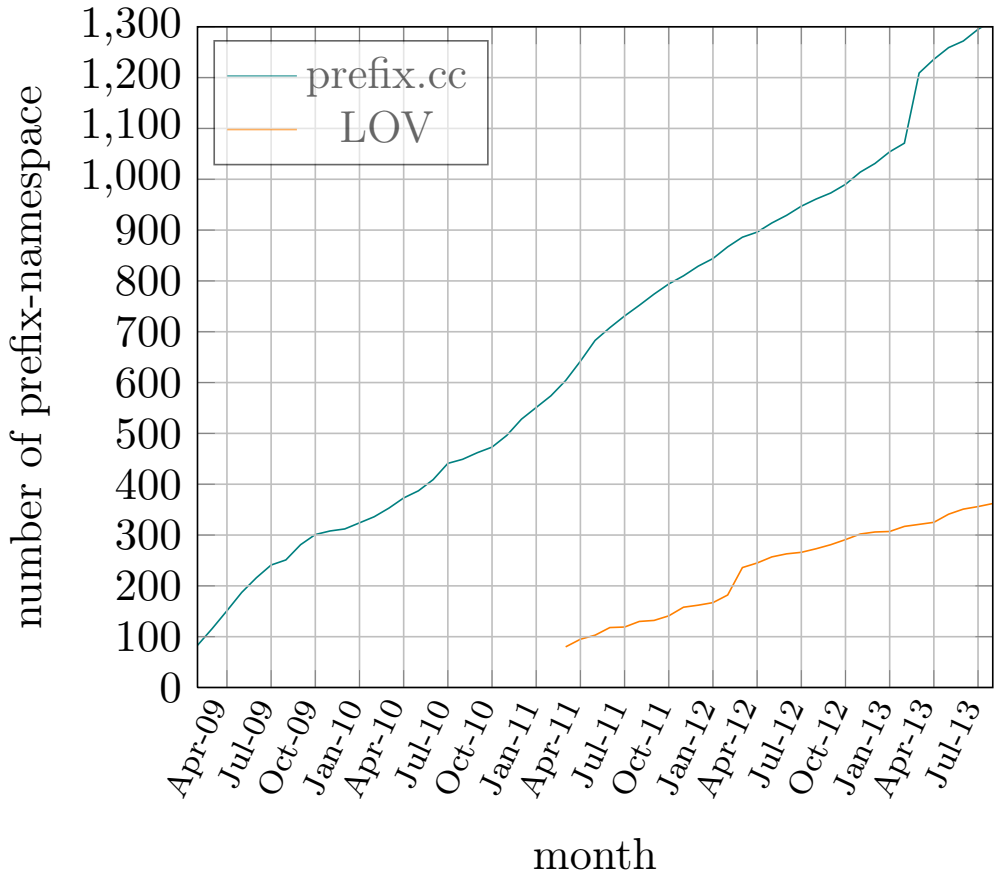


Figure 6.6: Evolution of the number of prefix-namespace pairs registered in prefix.cc and LOV

```
    }
  }
  FILTER (NOT EXISTS { [] vann:preferredNamespaceUri ?lovURI })
  OPTIONAL {
    [] vann:preferredNamespacePrefix ?prefix;
      vann:preferredNamespaceUri ?pccURI;
  }
}
ORDER BY ?prefix
```

Type of issue	# Vocabularies	%
pccURI and lovURI redirect to same resource	8	26.67%
lovURI already in prefix.cc as secondary	7	23.3%
Real conflicts	6	20%
pccURI is 404	4	13.3%
pccURI is an obsolete version	3	10%
lovURI is 404	1	3.3%
lovURI is an obsolete version	1	3.3%

Table 6.4: Type of issues encountered for vocabulary clashes

The first results[21] shown the following: $card(LOV) \bigcap card(PREFIX.cc) = 188$[22] and $card(LOV) - card(PREFIX.cc) = 133$[23] prefixes in LOV not yet registered in prefix.cc. At this point, a first batch of 80 prefixes/namespaces from LOV were safely imported in prefix.cc since there were no conflicts. For the remaining conflicting ones, they needed more in-depth analysis.

6.4.3 Second Task: Dealing with Conflicts between Prefix.cc and LOV

In the process of alignment, there were two types of conflicts and we provide appropriate actions and/or solutions accordingly:

- Clashes: cases where we have in both services the same prefix but different URIs;
- Disagreements on preferred namespace: cases where for the same URI, we found different prefixes.

Clashes.

We performed a SPARQL query as above to identify clashes in vocabularies (30). In Table 6.4, we identify seven different types of issues to deal with, such as (i) real conflicts, (ii) URIs are 404, (iii) URIs are obsolete versions and (iv) two URIs redirecting to the same resource.

Disagreements on namespace URIs.

The general idea is that if vocabulary editors have not included explicitly a `vann:preferredNamespacePrefix` in their description, the curators of LOV are free

[21] This query was performed in two weeks between March, 2nd and March, 20th 2013 and at this time, card(LOV) = 321 vocabularies while card(Prefix.cc) = 925

[22] `http://www.eurecom.fr/~atemezin/iswc2013/experiments/firstAlignments/intersection-prefixLOV-02-03.csv`

[23] `http://www.eurecom.fr/~atemezin/iswc2013/experiments/firstAlignments/inLovNotINPrefixcc-02-03.csv`

to change it and put whatever seems appropriate. At the same time, in prefix.cc, having multiple prefixes for the same namespace IRI in not a problem. However, we computed those prefixes in LOV that have different prefixes in prefix.cc. The following query retrieves the URIs falling in those disagreements:

```
PREFIX vann: <http://purl.org/vocab/vann/>
SELECT ?prefix ?lovURI ?prefixcc
FROM <http://prefix.cc/popular/all.file.vann> {
  SERVICE <http://lov.okfn.org/endpoint/lov> {
    SELECT ?prefix ?lovURI {
      [] vann:preferredNamespacePrefix ?prefix;
        vann:preferredNamespaceUri ?lovURI;
    }
  }
  FILTER (?pccURI = ?lovURI && ?prefix != ?prefixcc)
  OPTIONAL {
    [] vann:preferredNamespacePrefix ?prefixcc;
      vann:preferredNamespaceUri ?pccURI;
  }
}
ORDER BY ?prefix
```

From the results of this query (61 cases), we have three actions to perform:

- add the lovPrefix (prefix in LOV) in prefix.cc (e.g: adding `geod:http://vocab.lenka.no/geo-deling#`) to the existing `ngeoi` in *pccPrefix*.)
- add more alternative URIs to the existing prefix in prefix.cc (e.g: adding `prov:http://purl.org/net/provenance/ns#`) to the existing `hartigprov, prv` in *pccPrefix*)
- change a prefix in LOV[24] (e.g: lovPrefix `dc` for `http://purl.org/dc/terms` not in the list $\{dcterm, dcq, dct, dcterms\}$ has been replaced by `dce` in LOV).
- No changes when the lovPrefix is contained in the set of prefixes of prefix.cc.

6.4.4 Social Aspects

Several vocabularies are maintained by a community of users. As part of the alignment process, we contacted the authors, creators or maintainers (if they exist) of vocabularies to involve them as well in the process of changing prefixes, and agree with them to fix some issues regarding their vocabularies. From the homepages of the vocabulary authors and editors collected in LOV, we connect to their social

[24] `http://www.eurecom.fr/~atemezin/iswc2013/material/action-sameUriDifferentPrefixes.pdf`

platform accounts such as LinkedIn, Google+ or Twitter. Table 6.5 summarizes some cases of real conflicts where the LOV curators have to find and contact the editors of the vocabularies for negotiation.

prefix	lovURI	Remark
sp	http://data.lirmm.fr/ontologies/sp#	contact editor at LIRMM ($sp \Rightarrow osp$)
scot	http://scot-project.net/scot/ns#	contact editors at lovURI
media	http://purl.org/media#	contact editors for negotiation
pro	http://purl.org/spar/pro/	contact editors for negotiation
swp	http://www.w3.org/2004/03/trix/swp-1/	contact editors, fix on LOV side
wo	http://purl.org/ontology/wo/core#	contact editors
idemo	http://rdf.insee.fr/def/demo#	to resolve with INSEE

Table 6.5: LOV and prefix.cc conflicts resolution leading to contact vocabularies editors for negotiation. We provide the prefix, the URI in LOV and the action undertaken.

6.4.5 Finding Vocabularies in Prefix.cc

We want to find out in prefix.cc, which of the couples (prefix, URI) could be potentially a vocabulary to be further assess to be included in the LOV catalog. To address this question, we first compute all the differences on prefix.cc NOT in LOV, i.e. $PREFIX.CC$ MINUS ($LOV < INTERSECT > PREFIX.CC$), performing using a SPARQL query. This results in 742 URIs to be checked[25].

6.4.5.1 LOV Check API

We have implemented an API[26] that allows a user to run the LOV-Bot over a distant vocabulary. It takes as parameter the vocabulary URI to process and the time out (integer) specified to stop the process. The result of this action is a set of 26 property-values from which we are interested in using only 8 of them, namely:

- **uri** (string); uri of the vocabulary.
- **namespace** (string) ; namespace of the vocabulary.
- **prefix** (string) ; prefix of the vocabulary
- **inLOV** (boolean) ; indicates if the vocabulary is already in the Linked Open Vocabularies ecosystem.
- **nbClasses** (int) ; Number of classes defined in the vocabulary namespace.
- **nbProperties** (int) ; Number of properties defined in the vocabulary namespace.

[25] `http://www.eurecom.fr/~atemezin/iswc2013/experiments/input/notInLOV.json`
[26] `http://lov.okfn.org/dataset/lov/apidoc/`

- **dateIssued** (string) ; Vocabulary date of issue.
- **title** (Taxonomy) ; List of titles with language information if available.

The code below gives the response of our algorithm for the vocabulary identified at `http://ns.aksw.org/Evolution/`.

```
[caption={Sample output of a response of the Check API}]
{
 "dateIssued": "None",
 "inLOV": false,
 "namespace": "http://www.agfa.com/w3c/2009/clinicalProcedure#",
 "nbClasses": 47,
 "nbProperties": 29,
 "pccURI": "http://www.agfa.com/w3c/2009/clinicalProcedure",
 "prefix": "clinproc",
 "title": [
   {
     "dataType": null,
     "language": "en",
     "value": "Clinical Procedure"
   }
 ],
 "uri": "http://www.agfa.com/w3c/2009/clinicalProcedure"
},
```

6.4.5.2 Experiments

We wrote a script calling the LOV Check API on the URIs in prefix.cc for determining the candidates vocabularies to be inserted in LOV using the algorithm in Listing 2. We ran four times the experiments (possibly due to some network instabilities) in order to determine from which results what should be assessed. Table 6.6 gives an overview of the number of URIs with respectively the attribute "inLOV=false"(TP), "inLOV=true"(FP) and the errors occurred (Null returned, http/proxy or time out reached by the API).
Regarding the experiments, `Experiment4` gives stable results with less network errors. Therefore, we stick on this experiment to report our findings and analysis. We found that 227 (43.48%) are vocabularies in the sense of LOV since they have at least one property or one class. 297 vocabularies (56.51%) might have some problems (or are even not vocabularies at all) as they have neither classes nor properties. Regarding the presence of prefixes, we found 140 (61.67%) of them. The 227 vocabularies could all be inserted in the LOV catalog since they fulfill the current requirements of what is a " LOV-able vocabulary". In this list, we found vocabularies such as `rdf, rdfs, owl` that are used to build other vocabularies but are not yet integrated in the LOV catalog.

	TP(inLOV=false)	FP(inLOV=true)	Errors
Experiment1	525	44	173
Experiment2	403	26	313
Experiment3	351	28	363
Experiment4	522	44	176

Table 6.6: Experiments looking for stable results of finding vocabularies in prefix.cc.

Algorithm 2 finding vocabularies NOT in LOV from prefix.cc algorithm

Open $notInLOV.json$ file containing the prefix.cc URIs NOT in LOV
initialize $item$ as List
Initialize $result$ as collection of $item$
for each $pccURI \in notInLOVfile$ **do**
  $uri \leftarrow$ value of $pccURI$
  $uriv \leftarrow$ construct-valid uri
  call LOV-Check API with parameter $uriv$
  try/catch HTTPError, URLError, IOError, ValueError
  while no error raised **do**
    initialize $item$ to an empty List
    append $pccURI, prefix, inLOV, namespace, title, dateIssued, nbClasses, nbProperties$ in $item$ List
    append $item$ to $result$
  end while
end for
RETURN $output - result$

From the list of URIs that were not LOV-able vocabularies, we wanted to do more analysis by checking the RDF files using the Triple-Checker tool. Our aim is to be sure if we did not leave out some candidate vocabularies or if there are other type of errors such as parsing errors. Table 6.7 provides results classified into 4 categories:

- General errors such as loading files or proxy errors: 78.30%
- Candidate LOV-able vocabularies: 12.20%
- Clearly not vocabularies (`nbClasses` = `nbProperties` = 0), typically instances, datasets, html pages: 6.45%
- Others (mainly parsing errors): 3.05%

6.5 Vocabulary Ranking metrics

The linked data principles have gained significant momentum over the last few years as a best practice for sharing and publishing structured data on the Semantic

Total URIs	295	100%
Loading/404 errors	182	61.69%
Vocabularies	36	12.20%
Proxy errors	27	9.15%
50x, 40x errors	22	7.45%
Parsing errors	9	3.05%
Web Pages containers	9	3.05%
No triples found	8	2.71%
RDF data	2	0.67%

Table 6.7: Analysis of the URIs with no classes and no properties while using the LOV-Bot API

Web [83]. Before being published, data is modeled and ontologies or vocabularies are one of the key elements of a dataset. Vocabularies are the artefact that bring semantics to raw data. One of the major barriers to the deployment of linked data is the difficulty for data publishers to determine which vocabularies should be used since developing new vocabularies has a cost. Catalogues of ontologies are therefore a useful resource for searching terms (classes and properties) defined in those vocabularies. The Linked Open Vocabulary (LOV) initiative [37] is playing a significant role in providing such services to users who can search within curated vocabularies, fostering ontologies reuse. LOV focuses only on vocabularies submitted by users, which are then reviewed and validated by curators. In addition, LOV computes dependencies between vocabularies, keeps track of different versions of them in order to enable their temporal evolution.

To the best of our knowledge, recommending vocabularies to reuse are limited to "popular" or "well-known" ones. This paper proposes a metric combining different features such as how vocabularies are interlinked, or how they are used in real world datasets. This contribution originates also in the desire to bring the traditional concept of Information Content (IC) into the field of the semantic web applied to vocabularies. Many catalogs of ontologies already provide some ranking metrics based on some features. However, we are interested in applying the principles of IC on vocabularies to investigate if such techniques can give more insights in ontology ranking and ontology usage (e.g in visualization applications).

This Section is organized as follows: Section 6.5.1 defines the theory of Information Content, and the features used for applying Partition Information Content to vocabularies. We present our experiments on the LOV catalogue in the Section 6.5.4. We discuss how this ranking metric can be used for vocabulary design and maintenance in Section 6.5.5. We compare our results with other rankings for vocabularies in Section 6.5.6 before concluding and outlining future work (Section **??**).

6.5.1 Information Content Metrics

Based on probability theory, Information Content (IC) is computed as a measure of generated amount of surprise [87]. More common terms in a given corpus with higher chance of occurrence cause less surprise and accordingly carry less information, whereas infrequent ones are more informative. We reuse the notion of informativeness as the value of information associated with a given entity, where Information Content has a negative relation with its probability. The concept of Information Content can be used to rank each entity, term, or alphabet in the corpus. We apply the Partitioned Information Content to measure the informativeness of Linked Open Vocabularies as a semantic network of resources connected together using different range of relations, as described in [88]. Partitioned Information Content (PIC) is derived from the IC value using some weights. We empirically set those weights according to three features:

- (i) datasets using the vocabulary ($weight = 2$);
- (ii) *outlinks* from a vocabulary, i.e. whether a vocabulary reused other vocabularies ($weight = 1$);
- (iii) *inlinks* to a vocabulary, i.e. whether other vocabularies are reusing this vocabulary ($weight = 3$).

6.5.2 Information Content in Linked Open Vocabularies

This experiment aims at bringing the concept of informativeness in the field of terms semantically related as it is the case within semantic web ontologies. The ranking obtained can give additional information based on the Information Content theory to help reusing terms and detecting the ones that are less popular. This can then be used by applications consuming datasets described with these vocabularies. The equation (1) gives the formula for computing the IC value of a term (class or property):

$$IC(t) = -\log_2(\frac{\varphi(t)}{N}), \tag{6.1}$$

where N is set to be the maximum value corresponding to the term occurrence in the LOV aggregator (as of June 2014, this value is 3958, and it corresponds to the popularity of the `skos:prefLabel` property); and $\varphi(t)$ is the occurrence of the term (but not its popularity).
For computing $\varphi(t)$, we use two types of SPARQL queries depending on whether the term is a class (Listing 6.1) or a property (Listing 6.2 considers `owl:ObjectProperty`, `owl:DatatypeProperty` and `rdfs:Property`). Note that we do not yet take into account the `owl:equivalentClass` and `owl:equivalentProperty` axioms that may appear in some vocabularies. We leave this as a future work.

```
1 SELECT (count(?uri1) as ?occ)
2 WHERE {
3   ?uri1  ?p %%classURI . }
```

Listing 6.1: SPARQL query for computing the occurrence of a class

```
1 SELECT (count(?uri1) as ?occ)
2 WHERE {
3   ?uri1 +objectURI+ ?uri2.
4   FILTER  (?uri1 != ?uri2) }
```

Listing 6.2: SPARQL query for computing the occurrence of a property

6.5.3 Ranking Vocabularies using Information Content

For computing the PIC value, we use the following formula:

$$PIC(f) = w_f \times \sum_{i=1}^{n} IC(t_i), \tag{6.2}$$

where w_f is the weight related to vocabulary f.
We consider very important that a vocabulary is being reused by other vocabularies and implemented within real world datasets. For example, the `foaf` ontology is weighted 6 because it reuses vocabularies (1), it has been used in some datasets (2) and it is being reused by other vocabularies (3). The `dul`[27] vocabulary is weighted 3 because it doesn't reuse any vocabulary but it is instead used by several other vocabularies.

6.5.4 Experiments on Vocabularies

We use the LOV catalogue, and particularly the LOV aggregator[28] to look at the terms (classes and properties) to compute their Information Content (IC). LOV defines the *LOV Distribution* as the number of vocabularies in LOV that refer to a particular element and the *LOV popularity* as the number of other vocabulary elements that refers to a particular one. Based on the concept of Partitioned Information Content, we implement our ranking measure according to the algorithm 3. We take the subset of classes and/or properties with LOV popularity and LOV distribution greater than one. The initial set of vocabularies in LOV is 366. After filtering the candidate terms, we came out with a set of 161 vocabularies (44% or 161 vocabularies) for computing their ranking.
The Table 6.8 gives the Top 15-ranking of the vocabularies according to the informativeness of the classes and properties used within the LOV ecosystem. As the function is proportional to the number of terms, we use a threshold of 22 terms in

[27] `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl`
[28] `http://lov.okfn.org/endpoint/lov_aggregator`

the vocabularies. For example, the PIC value of `dcterms` is higher than `foaf`'s because the former uses 53 terms (39 properties and 14 classes), while the latter only 35 terms (9 classes and 26 properties), although they both have the same weight value.
The Table 6.9 shows the Top 20 namespaces of vocabularies according to the informativeness of the classes and properties used within the LOV ecosystem, along with their Information Content Value.

Rank	Prefix	PIC score
1	dcterms	1724.844
2	schema	1588.700
3	gr	1261.101
4	foaf	1033.197
5	bibo	876.205
6	time	816.2020
7	skos	805.287
8	dul	797.328
9	ptop	773.167
10	rdafrbr	640.834
11	vaem	630.621
12	ma-ont	508,694
13	prov	497.524
14	swrc	437.394
15	dce	428.618

Table 6.8: Top 15 vocabularies according to their PIC. All the prefixes used for the vocabularies are the ones used by LOV

6.5.5 Application of Information Content on Vocabularies

We foresee various applications using the ranking method based on the Information Content metric while designing semantic web applications, vocabulary life-cycle management or novel recommendation services. We make the following recommendations when using the PIC ranking method on vocabularies:

- Vocabularies on the Top PIC-ranking can be used in visualization applications, i.e. to be displayed to the user as much as possible.
- Terms with lower IC can be used in facetted browsing, and they seem appropriate for generating `sameAs` links during the interconnection and enrichment process. They might also be used for promoting the reuse of terms in vocabularies in general.
- The PIC-ranking could help the ontology designers to monitor and to assess the usage of some terms and lead to update the ontology accordingly. For example,

Rank	vocab term	IC value
1	skos:example	7.7806
2	dce:contributor	4.674
3	skos:scopeNote	4.365
4	dcterms:source	4.299
5	mads:code	3.922
6	mads:authoritativeLabel	3.922
7	vs:userdocs	3.847
8	dce:title	3.79
9	skos:hasTopConcept	3.4547
10	dce:description	2.758
11	dcterms:issued	2.553
12	dce:creator	2.518
13	skos:inScheme	2.202
14	skos:notation	1.924
15	dcterms:description	1.646
16	coll:List	0.761
17	vs:term_status	0.735
18	skos:definition	0.43
19	skos:prefLabel	0.009
20	foaf:Person	0

Table 6.9: Ranking of Top 20 terms (classes and properties) according to their IC value

Algorithm 3 Ranking vocabularies algorithm

```
REQUIRE Dump of lovaggregator file
Upload in a triple store for querying
Select subset of candidate vocabs LOVaggregatorendpoint
for term ∈ lovaggregator do
    if (LOVdistribution ≥ 1 )(LOVPopularity ≥ 1) then
        candidateterms ← append term
    end if
end for
for each term ∈ candidateterms do
    GROUP BY vocabulary namespace
    COMPUTE weight for each vocabulary
end for
INITIALIZE PICvector AS a vector
for each term ∈ candidateterms do
    while term ∈ vocabularySpace do
        ICterm ← function IC(term, vocabPrefix)
        ICvocab ← ∑ ICterm
    end while
    PICvocab ← weight(vocab)×ICvocab
    PICvector ← append (PICvocab)
    ORDER PICvector
end for
RETURN PICvector
```

it can be useful in extending the use of the properties such as `vs:term_status` or `owl:deprecated`.

- Such a ranking can be used to rank organizations or publishers of vocabularies in a time period (e.g. annual) as a way to encourage good qualities vocabularies and/or datasets on the cloud.

The use of the information content on LOV vocabularies can be applied in the datasets interlinking task and visualization applications workflow. For interlinking datasets, this method can help detecting properties with a lower PIC which will be a candidate for the interlinking tool. The PIC score can further be used to track the vocabularies terms status (i.e. `vs:term_status`) or `owl:deprecated` properties by dataset maintainers. From the list of namespaces having deprecated terms (Table 6.10), we observe some correlations with the PIC rank for the vocabularies `dcat (8), vcard (36), gr (6), wl (2), pav (1)` and `bibo (1)`[29]. More precisely, the presence of `gr` and `bibo` provides evidence of such a correlation, while the presence of `dcat` and `card` can be explained by the fact that those two vocabularies

[29] As of June 2014, there are 60 terms deprecated in LOV with the query `http://bit.ly/1aqcDf3`

are in a review process at W3C and subject to re-modeling respectively. Table 6.10 gives an overview of some namespaces with their deprecated terms.

6.5.6 Related Work and Discussion

In this section, we look at three other catalogues providing rankings for vocabularies: vocab.cc, LODStats and prefix.cc. vocab.cc[30] does not provide a ranking for vocabularies but rather proposes a rank for classes and properties. The proposed ranking presented in Table 6.11 is taken from the ranking of classes assuming the namespace is used only once per class.

prefix	#DeprecatedTerms	dcterms:modified
vcard	36	2013-09-25
dcat	8	2013-09-20
gr	6	2011-10-01
wl	2	2013-05-30
pav	1	2013-08-30
bibo	1	2009-11-04

Table 6.10: Sample of vocabularies with terms deprecated in LOV

Rank	**LOV-PIC**	**prefix.cc**	**vocab.cc**	**lodstats**
1	dcterms	yago	intervals	rdf
2	schema	rdf	foaf	rdfs
3	gr	foaf	time	owl
4	foaf	dbp	qb	dcterms
5	bibo	dce	scovo	skos
6	time	owl	freebase	foaf
7	skos	rdfs	mo	dce
8	dul	dbo	owl	void
9	ptop	rss	metalex	geo
10	rdafrbr	skos	doap	aktors
11	vaem	gldp	prov	ro
12	ma-ont	geo	void	obo
13	prov	sc	frbr	app
14	swrc	fb	skos	repo
15	dce	gn	dcterms	time

Table 6.11: Comparing ranking position when using PIC in LOV with respect to prefix.cc and vocab.cc

The LODStats ranking is focused on covering the number of datasets reused in the linked open data cloud [22], which is partially taken into account in our approach.

[30] `http://vocab.cc/v/tco`

The evidence of that is the first three vocabularies used (`RDF, RDFS, OWL`) which are considered as the meta model for designing vocabularies. Those vocabularies are not included into the LOV catalog and they do not appear in our ranking. The relative stable position of `foaf` in the four columns of the table suggests that there are equal popular terms. In addition, two other vocabularies have "relative" similar ranking using PIC and LODStats: `skos` and `dcterms`. Regardless the metric used, a short list of the "most popular vocabularies" based on their presence in the Top-15 of the four catalogues is: `foaf, skos` followed by `dcterms, time, dce, prov`.
Closer to our work, Schaible *et al.* reported on an empirical study involving 75 linked data experts and practitioners assessing reuse strategies based on various ranking decisions [89]. The goal is to find objective criteria for choosing which vocabularies to reuse and how many can be combined. LODStats and LOV are used to obtain the number of datasets using a specific vocabulary while *vocab.cc* is used for getting the number of occurrence of a vocabulary term. We propose a different metric to rank existing vocabularies that can be furthermore added as a new feature in such a study. One drawback in the model is to use the same weight for two vocabularies with different number of datasets reused. This could be address in the future by using a "function based" weighting for datasets reused (e.g. inverse logarithm) for computing the PIC score.

6.5.7 Summary

We have presented in this section a different perspective of ranking vocabularies using the principles of Information Content. By applying this concept to Linked Open Vocabularies, we tried to use features that we consider "relevant" to be taken into account when comparing vocabularies (e.g: datasets reused, external vocabularies). We compare with other rankings that are mostly based on the "popularity" of vocabularies. This work can path the way for assessing vocabularies with applications in a more systemic approach for recommending classes/properties in ontology management, or in visualization applications to propose the most *"oh yeah?"* suitable property to be visualized for RDF entities when there is large a large number of properties.

6.6 Datalift module for selecting vocabularies

Datalift platform comes with a module to map data objects and properties to ontology classes and predicates available in the LOV catalogue. Data2Ontology takes an input a "raw RDF", that is a dataset that has been converted directly from legacy format to triples. The goal is to help to publishers reusing existing ontologies for converting their dataset for easy discovery and interlinking. It consists of three main components assisting the publisher in selecting properties suitable for the dataset to be published.

⊙ **LOV component:** This component is in charge to connect with the LOV cata-

logue to retrieve up-to-date ontologies using the LOV search API[31].

⊙ **Matching Workflow:** Data2Ontology offers to map the data to LOV by automatically proposing a list of best matches. The suggestions are based on the algorithm that:

1. maximizes linguistic proximity between data properties and ontology predicates,
2. maximizes the estimated quality of the target ontology using LOV evaluation criteria, and
3. minimizes the number of candidates target ontologies to increase the overall semantic of the transformed dataset.

⊙ **SPARQL Generator:** This module receives as input the desired mappings and creates the SPARQL CONSTRUCT query needed to implement the mapping. The query can further be modified before the execution to generate a new dataset in the lifting process with Datalift.

Figure 6.7 illustrates the process of matching the properties with ontologies. Depending on the data properties and object properties used to map the elements of the datasets, Data2Ontology can also automatically infer the class model. However, the user can still distribute the predicate matches among several interconnected classes, as well as to add new predicates and values. Once the properties are matched with the desired predicates and their classes, the resulting model can be viewed as a graph.

6.7 Licenses Compatibility Checker

6.7.1 Background

The license of a dataset in the Web of Data can be specified within the data, or outside of it, for example in a separate document linking the data. In line with the Web of Data philosophy [90], licenses for such datasets should be specified in RDF, for instance through the Dublin Core vocabulary[32]. Despite such guidelines, still a lot of effort is needed to enhance the association of licenses to data on the Web, and to process licensed material in an automated way. The scenario becomes even more complex when another essential component in the Web of Data is taken into account: the vocabularies. Our goal is to support the data provider in assigning a license to her data, and verifying its compatibility with the licenses associated to the adopted vocabularies.
We answer this question by proposing an online framework called LIVE[33] (LIcenses VErification) that exploits the formal approach to licenses composition proposed

[31] `http://lov.okfn.org/dataset/lov/apidoc/#lov2search`
[32] `http://purl.org/dc/terms/license`
[33] The online tool is available at `http://www.eurecom.fr/~atemezin/licenseChecker/`

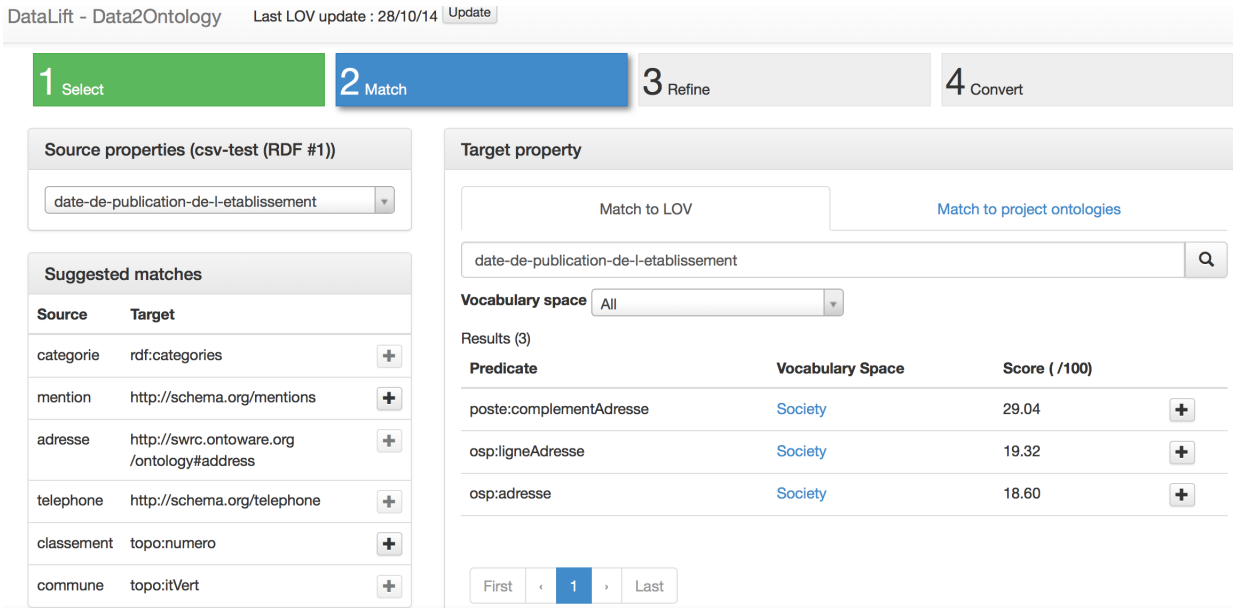


Figure 6.7: Matching data properties with ontology predicates in Data2Ontology module

in [91] to verify the compatibility of a set of heterogeneous licenses. LIVE, after retrieving the licenses associated to the vocabularies used in the dataset under analysis, supports data providers in verifying whether the license assigned to the dataset is compatible with those of the vocabularies, and returns a warning when this is not the case.

6.7.2 Statistics about licensed vocabularies

The first step to be addressed consists in analyzing how many vocabularies are licensed, and what is the distribution of such licenses. To achieve this goal, we started our analysis on the Linked Open Vocabularies repository. The LOV initiative stands as an observatory for the re-usable linked vocabularies ecosystem. The initiative goes beyond collecting and highlighting vocabulary metadata, and it now plays a major social role in promoting good practice and improving overall ecosystem quality of publishing vocabularies[34]. We crawled the LOV repository together with the LODstats one searching for licensed vocabularies. The results we obtained are as follows:

Licensed vocabularies : we have considered in total 419 vocabularies. The licensed vocabularies are 64 out of 419, eating about 16% of the the total number of considered vocabularies. The properties used to specify the licenses in the vocabularies are `http://creativecommons.org/ns#license` from the

[34]For more details about LOV, see `http://ercim-news.ercim.eu/en96/special/linked-open-vocabularies`

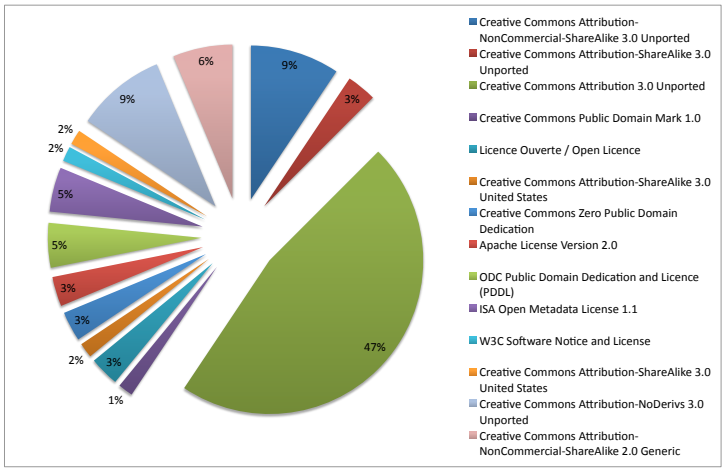


Figure 6.8: Licenses distribution in the LOV licensed vocabularies.

Creative Commons vocabulary [35] and `http://purl.org/dc/terms/license` from the Dublin Core vocabulary[36]. Even if the number of licensed vocabularies available on LOV and LODstats is rather low, analyzing the edits of the licensing information it is possible to note an increasing interest in providing further metadata about the data published on the Web of Data, and it holds also for vocabularies[37]. Another interesting result shows that 4 out of 13 vocabularies retrieved searching for the "top 20 most used entities in the LOD cloud[38]" has an explicit license associated. These vocabularies are Good Relations[39], DBpedia, GeoNames[40], and FRBR[41].

Licenses distribution : the distribution of the licenses in the licensed vocabularies we retrieved is visualized in Figure 6.8. The most adopted license is Creative Commons Attribution (CC-BY) (30 out of 64 licensed vocabularies), and Creative Commons licenses in general represent the 85% of the licenses used for vocabularies licensing, confirming the trend shown for licensed datasets [92, 91]. Another popular license is ODC Public Domain Dedication and License (PDDL)[42] followed by the W3C Software Notice and License[43].

We are aware that the obtained results are referred to the data available on LOV and LODstats, and that such repositories of vocabularies are not exhaustive. However,

[35] `http://creativecommons.org/ns`
[36] `http://dublincore.org/documents/dcmi-terms/`
[37] In this work, we consider vocabularies as data. Other interpretation of the role of vocabularies are discussed in the conclusions.
[38] `http://lod-cloud.net/`
[39] `http://purl.org/goodrelations/v1#`
[40] `http://www.geonames.org/ontology#`
[41] `http://vocab.org/frbr/core.html#`
[42] `http://opendatacommons.org/licenses/pddl/1.0/`
[43] `http://bit.ly/W3C-license`

they provide us a reliable picture of what is the ongoing trend in licensing vocabularies[44]. This is particularly true in the case of LOV that furthermore supports a number of good practices for the publication of a vocabulary, among which the addition of the license associated to the vocabulary is highly encouraged.

6.7.3 Related work about licenses in the Web of Data

In the Web scenario, a number of works address the problem of representing and/or reasoning over licensing information. Iannella [45] presents the Open Digital Rights Language (ODRL) for expressing rights information over content, and Gangadharan et al. [93] further extend ODRL developing the ODRL-S language to implement the clauses of service licensing. Gangadharan et al. [94] address the issue of service license composition and compatibility analysis basing on ODRL-S. They specify a matchmaking algorithm which verifies whether two service licenses are compatible. In case of a positive answer, the services can be composed and the framework determines the license of the composite service. Nadah et al. [95] propose to assist licensors' work by providing them a generic way to instantiate licenses, independent from specific formats, and then they translate the license expressed in generic terms into more specific terms compliant with the specific standards used by distribution systems, i.e., ODRL and MPEG Rights Data Dictionaries. Truong et al. [96] address the issue of analyzing data contracts, based on ODRL-S again. Contract analysis leads to the definition of a contract composition where first the comparable contractual terms from the different data contracts are retrieved, and second an evaluation of the new contractual terms for the data mash-up is addressed. Krotzsch and Speiser [97] present a semantic framework for evaluating ShareAlike recursive statements. In particular, they develop a general policy modelling language, then instantiated with OWL DL and Datalog, for supporting self-referential policies as expressed by CC. Gordon [98] presents a legal prototype for analyzing open source licenses compatibility using the Carneades argumentation system. Finally, Rodiguez-Doncel et al. [99, 100] discuss licenses patterns for Linked Data. In particular, they first analyze and discuss six rights expression languages, abstracting their commonalities and outlining their underlying pattern. Second, they propose the License Linked Data Resources pattern which provides a solution to describe existing licenses and rights expressions both for open and not open scenarios. All these works either propose new ways to model licenses information or new formal frameworks to deal with rights. In this paper, we do not address none of these issues, and we adopt the formal framework proposed in [91]. Also Pucella and Weissman [101] propose a logic to check whether the user's actions follow the licenses' specifications. However, as they do not deal with compatibility, do not provide a deontic account of licenses' conclusions, and their logic is not able to handle conflicting licenses, we choose and adapt the deontic logic of [91], which better suits our needs.

[44] We refer the reader interested into statistics about the distribution of licenses on the Web of Data to [92, 91].

[45] `http://odrl.net/1.1/ODRL-11.pdf`

Up to our knowledge, the issue of licensed vocabularies has never been addressed. More precisely, no available framework exists dealing with such licenses and verifying in an automated way their potential compatibility with the license associated to datasets. The goal of supporting users in dealing with licensing information has been recently addressed by Cabrio et al. [102] with a different goal, i.e., supporting data publishers in creating RDF licenses representations from natural language texts.

6.7.4 The LIVE Framework

The LIVE framework is a Javascript application, combining HTML and Bootstrap. Hence, installation has no prerequisite. Since the tool is written in Javascript, the best way to monitor the execution time is with the `performance.now()` function. We use the 10 LOD datasets with the highest number of links towards other LOD datasets available at `http://lod-cloud.net/state/#links`. For each of the URLs in Datahub, we retrieve the VoID[46] file in Turtle format, and we use the *voidChecker* function[47] of the LIVE tool to retrieve the associated license, if any. The goal of the LIVE framework is to support data producers to assign a license to the data ensuring the consistency of such license with respect to the licenses assigned to the vocabularies she exploits in the dataset. The input of the LIVE framework (Figure 6.9) consists in the dataset (URI or VOiD) whose license has to be verified. The framework is composed by two modules. The first module takes care of retrieving the vocabularies used in the dataset, and for each vocabulary, retrieves the associate license[48] (if any) querying the LOV repository. The module searches out also the license associated to the dataset itself. When all the licensing information of interest has been obtained, the module provides such set of licenses to the compatibility checking module. The second module takes as input the set of licenses (i.e., the licenses of the vocabularies used in the dataset as well as the license assigned to the dataset) to verify whether they are compatible with each others. The result returned by the module is a *yes/no* answer. In case of negative answer, the data provider is invited to change the license associated to the dataset and check back again with the LIVE framework whether further inconsistencies arise.

6.7.5 Licensing information from vocabularies and datasets.

Two use-cases are taken into account: a SPARQL endpoint, or a VoID file in Turtle syntax. In the first use case, the tool retrieves the named graphs present in the repository, and then the user is asked to select the URI of the graph that needs to be checked. Having that information, a SPARQL query is triggered, looking for entities declared as `owl:Ontology`, `voaf:Vocabulary` or object of the `void:vocabulary` property. The final step is to look up the LOV catalogue to check whether they declare any license. There are two options for checking the license: *(i)* a *"strict*

[46] `http://www.w3.org/TR/void/`

[47] `http://www.eurecom.fr/~atemezin/licenseChecker/voidChecker.html`

[48] Note that the LIVE framework relies on the dataset of machine-readable licenses (RDF, Turtle syntax) presented in [102].

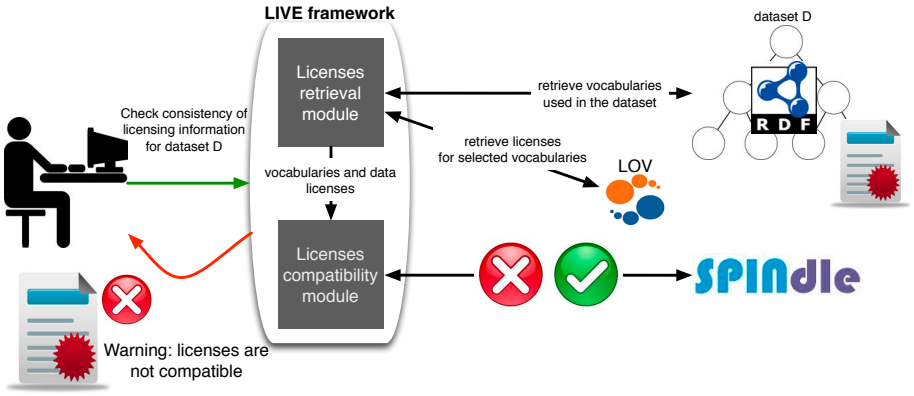


Figure 6.9: LIVE framework architecture.

checking" where the `FILTER` clause contains exactly the namespace of the submitted vocabulary, or *(ii)* a *"domain checking"*, where only the domain of the vocabulary is used in the `FILTER` clause. This latter option is recommended in case only one vocabulary has to be checked for the license. In the second use case, the module parses a VoID file using a N3 parser for Javascript[49], and then collects the declared vocabularies in the file, querying again LOV[50] to check their licensing information. When the URIs of the licenses associated to the vocabularies and the dataset are retrieved, the module retrieves the machine-readable description of the licenses in the dataset of licenses [102]. More specifically, such dataset is composed by 37 licenses, comprising all the licenses adopted to certify data in the Linked Data cloud (as all the Creative Commons licenses[51]), software licenses (as Mozilla Public License[52] and Microsoft License[53]), and additional licenses for other material on the Web (as the UK Open Government license, and the New Free Documentation License[54]). The dataset provides the licenses in RDF using the Turtle syntax, however Creative Commons licenses are also available in XML/RDF format on the CC website[55]. Figure 6.10 shows the user interface for quering a graph and sample results provided by LIVE tool.

6.7.6 Licenses compatibility verification.

The logic proposed in [91] and the licenses compatibility verification process has been implemented using SPINdle [103] – a defeasible logic reasoner capable of inferencing defeasible theories with hundredth of thousand rules.

[49] `https://github.com/RubenVerborgh/N3.js`

[50] Since LOV endpoint does not support the JSON format in the results, we have uploaded the data in `eventmedia.eurecom.fr/sparql`.

[51] `http://creativecommons.org/licenses/`

[52] `http://www.mozilla.org/MPL/2.0/`

[53] `http://referencesource.microsoft.com/referencesourcelicensing.aspx`

[54] `http://www.gnu.org/copyleft/fdl.html`

[55] For instance, Creative Commons Attribution 4.0 license is available at `http://creativecommons.org/licenses/by/4.0/rdf`

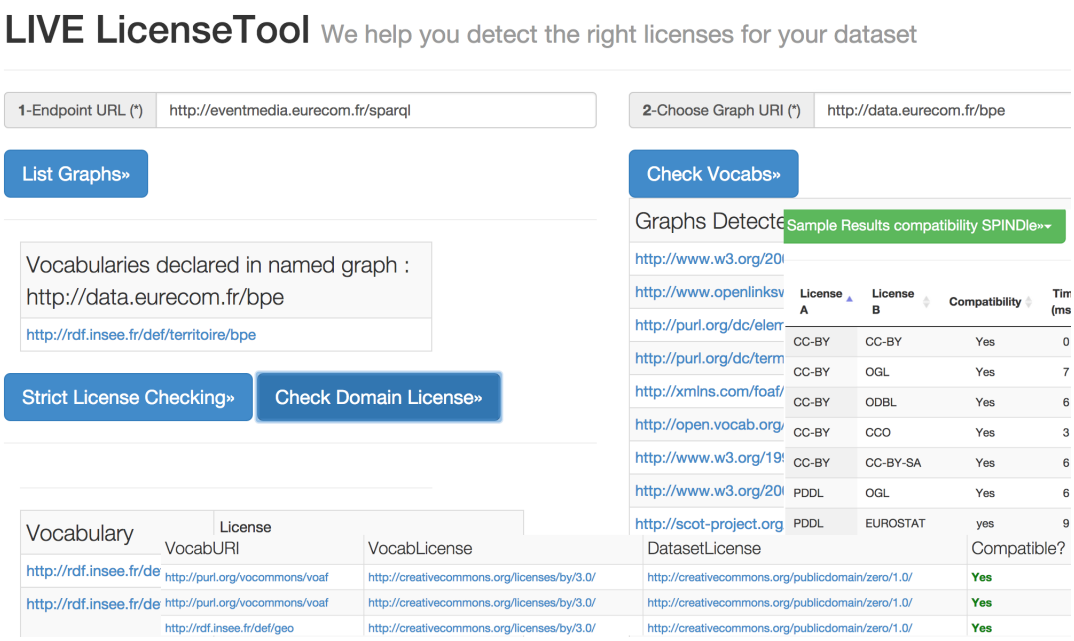


Figure 6.10: LIVE tool user interface and sample results

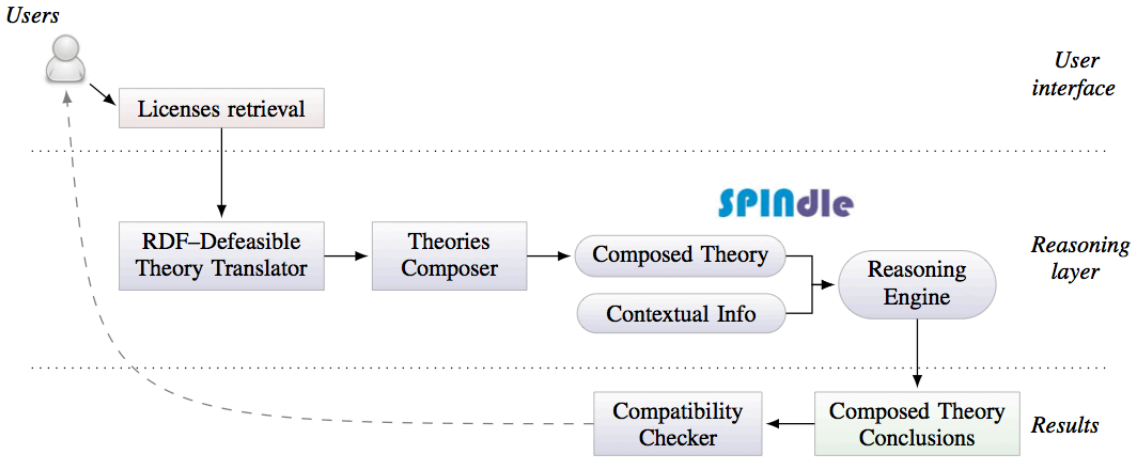


Figure 6.11: Licenses compatibility module.

As depicted in Figure 6.11, after receiving queries from users, the selected licenses (represented using RDF) will be translated into the DFL formalism supported by SPINdle using the *RDF-Defeasible Theory Translator*. That is, each RDF-triple will be translated into a defeasible rule based on the subsumption relation between the *subject* and *object* of a RDF-triples. In our case, we can use the subject and object of the RDF-triples as the antecedent and head of a defeasible rule, respectively. Besides, the translator also supports direct import from the Web and processing

of RDF data into SPINdle theories. The *RDF-Defeasible Theory Translator* will translate the RDF-licenses into the DFL formalism supported by SPINdle.
The translated defeasible theories will then be composed into a single defeasible theory based on the logic proposed in [91], using the *Theories Composer*. Afterwards, the composed theory, together with other contextual information (as defined by user), will be loaded into the SPINdle reasoner to perform a compatibility check before returning the results to the users.
We have evaluated the time performances of the LIVE framework in two steps (Table 6.12).

Dataset	**LicRetrieval(ms)**	**#vocabularies**	**LicCompatibility(ms)**	**LIVE(ms)**
rkb-explorer-dblp	4,499	1	0	4,499
rkb-explorer-southampton	14,693	1	0	14,693
rkb-explorer-eprints	3,220	1	0	3,220
rkb-explorer-acm	3,007	1	0	3,007
rkb-explorer-wiki	14,598	1	0	14,598
rkb-explorer-rae2001	3,343	1	0	3,343
rkb-explorer-citeseer	2,760	1	0	2,760
rkb-explorer-newcastle	3,354	1	0	3,354
rkb-explorer-kisti	4,094	5	6	4,100
270a.info	13,202	48	8	13,210

Table 6.12: Evaluation of the LIVE framework.

First, we evaluate the time performances of the licenses compatibility module: it needs about 6ms to compute the compatibility of two licenses. Second, we evaluate time performances (Chrome v. 34) of the whole LIVE framework for the 10 LOD datasets with the highest number of links towards other LOD datasets, considering both the licenses retrieval module and the licenses compatibility one. The results show that LIVE provides the compatibility evaluation in less than 5 seconds for 7 of the selected datasets. Time performances of LIVE are mostly affected by the first module while the compatibility module does not produce a significant overhead. For instance, consider Linked Dataspaces[56], a dataset where we retrieve the licensing information in both the dataset and the adopted vocabularies. In this case, LIVE retrieves in 13.20s 48 vocabularies, the license for the dataset is CC-BY, and the PDDL license is attached one of the vocabularies[57]. The time for verifying the compatibility is 8ms, leading to a total of 13.208s.

6.7.7 Future perspectives

We have introduced the LIVE framework for licenses compatibility. The goal of the framework is to verify the compatibility of the licenses associated to the vocabularies exploited to create a RDF dataset and the license associated to the dataset itself. Several points have to be taken into account as future work. More precisely, in the present paper we consider vocabularies as data but this is not the only possible

[56] `http://270a.info/`
[57] `http://purl.org/linked-data/cube`

interpretation. For instance, we may see vocabularies as a kind of compiler, such that, after the creation of the dataset then the external vocabularies are no more used. In this case, what is a suitable way of defining a compatibility verification? We will investigate this issue as well as we will evaluate the usability of the online LIVE tool to subsequently improve the user interface.

6.8 Summary

In this chapter, we have presented our contributions on achieving some of the guidelines of the best practices of publishing Linked Data, by presenting the LOV catalogue, the harmonization of LOV with other catalogues, the importance of using semantics to rank vocabularies. Besides, we have presented Data2Ontology, a module aim at helping the reuse of existing vocabularies during the publication of data in Datalift. We finished by presenting `LIVE`, an online tool to check the compatibility between datasets and vocabularies based on the RDF-defeasible of SPINdle.

Chapter 7

Conclusions and Future Perspectives

"How do we know that Semantic Web technologies were actually better here, as opposed to being what the developers found most familiar?"
(D. Karger)[1]

7.1 Conclusions

This thesis is focused on the challenges of publishing geodata on the Web and a more generic approach to visualize data as Linked Data target to lay-users. The former considers the diversity of different formats used to publish legacy geospatial data, the different projections and CRSs and the representation of complex geometries. The latter approach is different to the state-of-the-art in visualizations where the complexity of SPARQL and RDF is not sufficiently hidden to the users. A deep analysis of the literature has revealed some limitations in the publication of geospatial data and visualization tools, namely:

- A few presence of complex geometries exposed in structured representation, instead of literals.
- The absence of an explicit reference to CRSs in direct georeference data on the Web.
- Absence of visualization tool targeted to lay users to easily grasp the essence of the underlying data published as LOD.
- Many data silos for applications built and published on the Web, lost in many html pages.
- Few tools that provide an integrated environment for publishing raw data into Linked Data, from data modeling until the final step of storing the dataset in an endpoint.
- The difficulty for publishers to understand and check the compatibility of the licenses between vocabularies and datasets.

[1] `http://goo.gl/hQQ3h5`

In this thesis, we have provided different vocabularies that all together support the publication of geodata integrating almost all the CRSs, extending the existing vocabularies. The vocabularies have been used to publish the French Administrative Units, with the data compatible with GeoSPARQL standards. Regarding the visualizations, after reviewing visual tools and existing applications on the Web, we have developed an ontology to better expose the data on the Web for better interoperability. Besides we have proposed a framework to generate automatically visualizations based on categories detected on datasets published as Linked Data, based on predefined categories used in InfoVis and mapped with vocabularies.

7.1.1 Review of the Contributions

This section reviews the main contributions of this thesis and the solutions to solved some of the open research problems in publishing and consuming data on the Semantic Web:

- The model and implementation of a vocabulary for geometry, topological entities and CRSs.
- The implementation of an API for converting data between different CRSs accessible on the Web.
- We have contributed in the development of the Datalift platform, an integrated environment to publish raw data on the Web.
- The comparison of triple stores for geodata against the geometries handled (literal or structured) to assess which one to use when publishing geospatial data.
- The publication of the French Administrative Units available at `data.ign.fr` endpoint, based on the vocabularies developed and implemented. Moreover, we have provided interlinking with relevant existing geospatial datasets.
- We have published the CRSs with unique URIs for better look up and integration in structured geometries on the Web.
- The contribution to the *French LOD (FrLOD)* cloud, with more datasets published using the Datalift platform, and covering the French territory.
- We have proposed a generic approach to automatically generate visualizations based on predefined categories.
- We have developed two innovative applications consuming events and statistical datasets
- We have proposed a vocabulary and a tool that can improve the discovery of applications contests in Open Data events.

- Finally, we have proposed an approach to harmonize prefixes used in different catalogues of vocabulary.
- We have developed new ranking metrics for vocabularies based on Information Content theories.
- Finally, we have built a more efficient tool for checking licenses compatibility between vocabularies and datasets.

7.2 Future Perspectives

In this thesis, we have tackled some open research problems within the context of publishing and consuming open data on the Web but there are still open issues to resolve or extensions to implement. We would like to mention some of the most important from our perspectives, based on different tasks in the workflow of the publication.

7.2.1 Opportunities and Challenges for IGN-France

The need for interoperable reference geographic data to share and combine geo-referenced environmental spatial information is particularly acknowledged by the INSPIRE Directive. For geographic data producers, the benefit of publishing their data on the Web according to Linked Data (LD) principles is twofold. On the one hand, their data are interoperable with other published datasets and they can be referenced by external resources and used as spatial reference data, which would not have been straightforward when published according to spatial data infrastructures (SDI) standards. On the other hand, the use of semantic Web technologies can help addressing interoperability issues which are not solved yet by geographic information standards. Moreover, there are different types of license policies to access data at IGN (e.g., research purpose, commercial use, access on demand, etc.), with some of them not necessary "open" or free to access: (e.g., BD TOPO®). Although there is a clear understanding of the benefits of publishing and interconnecting data on the web, ongoing investigations on how to combine licenses on datasets are under consideration at IGN. Two solutions are under investigation: (i) different license policies attached to datasets and (ii) the use of a security access mechanism on top of the datasets granting access based on a predetermined configuration on named graphs and resources. According to Linked data principles URIs should remain stable, even if administrative units change or disappear. This implies adapting the data vocabulary in order to handle data versioning and real world evolutions. This issue will be addressed in a future work, as we plan to release a spatio-temporal dataset describing the evolution of communes since the French Revolution. Another issue deals with the automation of the whole publication process, from traditional geographic data to fully interconnected RDF data. The last issue deals with the use of multiple geometries for describing a geographic feature: geometries with different levels of detail, different CRS, different representation choices. This has been

superficially addressed in our use case with the use of both polygons and points for representing respectively the surface and the centroid of departments, but should be further investigated for both query answering and map design purposes.

7.2.2 Generic Visualizations on Linked Data

We plan to use a more exhaustive set of vocabularies in our generic queries for detecting those categories, plugging directly the wizard to the LOV catalogue. Regarding the aggregation properties, it can be extended to take into account other semantic relations (e.g: `skos:exactMatch`). Additionally, we plan to make an evaluation of the prototype and compare it to related tools such as the ones aiming to build a dataset profile. We also need to quantify when a category is "important" within a dataset. For example, is it enough for a dataset to be classified GEODATA with ten triples containing location ? From which number of triples found could be assign the categories and hence the visualizations? These issues can further be investigated to find the best trade-off. Another drawback of our work on visualizations is the lack of users' evaluation, with sound experiments to understand users' needs, more focusing on the semantic aspects than the Web. A natural follow-up is to go through this evaluations and re-adapt the applications/visualizations based on the results.

7.2.3 Vocabularies and LOV

Regarding the harmonization of prefixes, the work can be extended in several directions. Sticking to the two services we have studied and already contributed to harmonize, the possible next steps would be to automate as far as possible the tasks that have been made semi-automatically so far: *i)* developing a unique interface for submitting namespaces and prefixes to both services; *ii)* bridging the LOV back-office and the prefix-cc database using both services API in order to publish a list of common recommended prefixes. The latter goes beyond the limited framework of the two original services since such a list could be consolidated and endorsed by the main actors in vocabulary publication and management, and recommended for use in linked data applications. This could be picked up by the upcoming W3C Vocabulary Management Working Group as part of the new Data Activity[2].
As per ranking vocabularies, we aim to take into account the equivalence axioms (between classes and properties) when computing the Information Content, and more generally, all sort of semantic relationships between terms. Also, we plan to compare our ranking model with other ranking approaches such as graph-based ones (e.g., pagerank). Another future direction work is to investigate the dependency ranking between vocabularies, by focusing on a specific type of "inlinks" (i.e. extensions, generalization) and study how they affect the PIC values.
We have made the assumption in this thesis that the access to data was either by querying the SPARQL endpoint, or by browsing or by downloading the dumps.

[2] `http://www.w3.org/2013/05/odbp-charter.html`

Recently, it is emerging a new way of accessing the data on the Web: through the triple pattern fragments. Linked Data Fragments [**?**, RUBEN]ims at exploring interfaces to solve queries at the client side with server data. Servers can offer data at low processing cost in a way that enables client-side querying. Thus, moving intelligence from the server to the client. One possible direction of study could be to use this concept for evaluating endpoints consuming only structured geometries versus literal for real live applications. Finally, triple fragments concepts can be apply to detect also patterns for visualization in different endpoints.

Appendix A

Installation instructions for the JavaScript plugin

The system consists of three components, each having their own code repository:

- Admin-interface: This is the interface that event organizers use to create new events and to get the embed code for embedding the event on their own website. It is available at `https://github.com/EurecomApps4Eu/admin-interface`.
- REST-interface: This interface provides a RESTful service for events and applications, available at `https://github.com/EurecomApps4Eu/rest-interface`
- Embeddable script at `https://github.com/EurecomApps4Eu/event-website`. This repository contains the code that is embedded on the event organizer's own website. It will fetch the event and application information from the REST-interface, and then display the information directly on the event organizers webpage.

The three different components can be installed on different computers, if needed. REST-interface requires Node.js and MongoDB to be installed, whereas Admin-interface and Event-website produces static files (when running the build task) that can be hosted on any web server that is capable of serving static files.

A.1 Installing and configuring the REST-interface

```
1. Clone the repository
2. Install dependencies by running the command "npm install"
3. Configure the system by editing "config.js"-file
4. Start the service by running "node app.js {PORT}", and replace {PORT}
with the port you want to run the application in (typically port 80 for HTTP)
```

A.2 Installing and configuring the Admin-interface

```
1. Clone the repository
2. Install Node dependencies by running the command "npm install"
3. Install Bower dependencies by running the command "bower install"
(if bower is not installed run "npm install bower -g")
```

```
4. Configure the system by editing file "app/scripts/app.js".
Look for appSettings-part in the file.
5. Build the system with command "grunt build"
(if grunt is not installed, run "npm install grunt -g").
This will produce static HTML/CSS/JS-files that can be hosted on any web server.
```

A.3 Installing and configuring the Event-website

```
1. Clone the repository
2. Install Node dependencies by running the command "npm install"
3. Install Bower dependencies by running the command "bower install"
(if bower is not installed run "npm install bower -g")
4. Configure the system by editing file "config.js"
5. Build static HTML/CSS/JS-files by running "grunt all"
```

Bibliography

[1] Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman II. Five stars of linked data vocabulary use. *Semantic Web Journal*, 2014. `http://geog.ucsb.edu/~jano/swj653.pdf`. vii

[2] Maria del Carmen. Suárez-Figueroa. *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. PhD thesis, Universidad Politecnica de Madrid, Spain, June 2010. http://oa.upm.es/3879/. xvi, 130, 133

[3] Bernadette Hyland, Ghislain Atemezing, and Boris Villazon-Terrazas (eds). Best practices for publishing linked data. W3C Working Group Note, 2014. `http://www.w3.org/TR/ld-bp/`. xviii, 9, 121, 125

[4] Mathieu DÁquin and Natasha F. Noy. Where to publish and find ontologies? a survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11(0):96–111, 2012. xviii, 123, 127, 129, 131

[5] Tim Berners-Lee. Design issues for linked data, 2006. `http://www.w3.org/DesignIssues/LinkedData.html`. 1

[6] A. Jentzsch, R. Cyganiak, and C. Bizer. State of the lod cloud (september 2011), 2011. `http://lod-cloud.net/state/`. 1

[7] Schmachtenberg Max, Bizer Christian, and Paulheim Heiko. Adoption of the linked data best practices in different topical domains. In *Proceedings of the ISWC 2014, RDB Track (To appear)*, 2014. `http://dws.informatik.uni-mannheim.de/fileadmin/lehrstuehle/ki/pub/SchmachtenbergBizerPaulheim-AdoptionOfLinkedDataBestPractices.pdf`. 1

[8] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011. 1

[9] John Goodwin, Catherine Dolbear, and Glen Hart. Geographical linked data: The administrative geography of great britain on the semantic web. *Transactions in GIS*, 12:19–30, 2008. 3

[10] Sören Auer, Jens Lehmann, and Sebastian Hellmann. LinkedGeoData - Adding a Spatial Dimension to the Web of Data. In *8^{th} International Semantic Web Conference (ISWC'09)*, 2009. 3, 22

[11] Alexander de León, Luis M. Vilches, Boris Villazón-Terrazas, Freddy Priyatna, and Oscar Corcho. Geographical linked data: a Spanish use case. In *International Conference on Semantic Systems (I-SEMANTICS'10)*, Graz, Austria, 2010. 3, 22, 42

[12] Juan Salas and Andreas Harth. Finding spatial equivalences across multiple RDF datasets. In *4th International Terra Cognita Workshop*, pages 114–126, Bonn, Germany, 2011. 3, 21

[13] Max J Egenhofer. Toward the semantic geospatial web. In *Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 1–4. ACM, 2002. 3, 15

[14] Manolis Koubarakis, Manos Karpathiotakis, Kostis Kyzirakos, Charalampos Nikolaou, and Michael Sioutis. Data models and query languages for linked geospatial data. In *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, pages 290–328, 2012. http://dx.doi.org/10.1007/978-3-642-33158-9$_8$. 6, 48

[15] Richard Cyganiak and Dave Reynolds (eds). The rdf data cube vocabulary. W3C Recommendation, 2014. `http://www.w3.org/TR/vocab-data-cube/`. 9, 105, 107

[16] Dave Reynolds. The organization ontology. W3C Recommendation, 2014. `http://www.w3.org/TR/vocab-org/`. 9

[17] A. Peter Burrough and A. Rachael McDonnell. Principles of geographical information systems. pages 17–34, 1998. 16

[18] Ana-Maria OLTEANU. *Fusion de connaissances imparfaites pour láppariement de données géographiques*. PhD thesis, Universite Paris-Est, France, October 2008. 17

[19] ESRI. Esri shapefile technical description. An ESRI White Paper, 1998. `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`. 18

[20] Open Geospatial Consortium Inc. Simple feature access - part 1: Common architecture. Technical report, 2011. `http://www.opengeospatial.org/standards/sfa`. 18

[21] Howard Butler (Hobu Inc.), Martin Daly (Cadcorp), Allan Doyle (MIT), Sean Gillies (UNC-Chapel Hill), Tim Schaub (OpenGeo), and Christopher Schmidt (MetaCarta). The geojson format specification, 2008. `http://geojson.org/geojson-spec.html`. 18

[22] Jan Demter, Sören Auer, Michael Martin, and Jens Lehmann. LODStats – An Extensible Framework for High-performance Dataset Analytics. In *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 353–362, Galway, Ireland, 2012. 20, 148

[23] John Goodwin, Catherine Dolbear, and Glen Hart. Geographical Linked Data: The Administrative Geography of Great Britain on the Semantic Web. *Transactions in GIS*, 12:19–30, 2008. 22

[24] Matthew Perry and John Herring. OGC GeoSPARQL- A Geographic Query Language for RDF Data. In *OGC Implementation Standard, ref: OGC 11-052r4*, 2012. 22

[25] Auguste Ghislain Atemezing and Raphaël Troncy. Comparing Vocabularies for Representing Geographical Features and Their Geometry. In *5th International Terra Cognita Workshop*, Boston, USA, 2012. 25, 31

[26] INSPIRE Thematic WG CRS and Geographical Grid Systems . Guidelines INSPIRE Specification on Coordinate Reference Systems , 2009. `http://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_Specification_CRS_v3.0.pdf`. 26, 37

[27] International Organization for Standardization . ISO 19125-1, Geographic information- Simple feature access - Part 1: Common architecture, 2004. 35

[28] Kostas Patroumpas, Michalis Alexakis, Giorgos Giannopoulos, and Spiros Athanasiou. Triplegeo: an etl tool for transforming geospatial data into rdf triples. 2014. 42

[29] Jhonny Saavedra, Luis M. Vilches-Blázquez, and Alberto Boada. Cadastral data integration through linked data. In Granell (Eds): Connecting a Digital Europe through Location Huerta, Schade and Place., editors, *Proceedings of the AGILE'2014 International Conference on Geographic Information Science Castellón, June, 3-6*, 2014. `http://hdl.handle.net/10234/98742`. 42

[30] International Organization for Standardization . ISO 19152, Geographic information â Land Administration Domain Model (LADM), 2012. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51206`. 42

[31] Fayçal Hamdi, Nathalie Abadie, Bénédicte Bucher, and Abdelfettah Feliachi. Geom-rdf: A fine-grained structured representation of geometry in the web. In *Proceedings of the 1st International Workshop on Geospatial Linked Data, 1 September, Leipzig, Germany*, 2014. 43

[32] W3C SPARQL Working Group. Sparql query language for rdf. W3C Recommendation 21 March 2013, 2013. `http://www.w3.org/TR/sparql11-overview/`. 47

[33] Carlos Buil Aranda, Aidan Hogan, Jürgen Umbrich, and Pierre-Yves Vandenbussche. SPARQL web-querying infrastructure: Ready for action? In Alani et al. [104], pages 277–293. 47

[34] Robert Battle and Dave Kolas. Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web*, 3(4):355–370, 2012. 48, 49

[35] George Garbis, Kostis Kyzirakos, and Manolis Koubarakis. Geographica: A benchmark for geospatial RDF stores (long version). In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, pages 343–359, 2013. 48

[36] Kostis Kyzirakos, Manos Karpathiotakis, and Manolis Koubarakis. Strabon: a semantic geospatial dbms. In *The Semantic Web–ISWC 2012*, pages 295–311. Springer, 2012. 49

[37] François Scharffe, Ghislain Atemezing, Raphaël Troncy, Fabien Gandon, Serena Villata, Bénédicte Bucher, Fayçal Hamdi, Laurent Bihanic, Gabriel Képéklian, Franck Cotton, Jérôme Euzenat, Zhengjie Fan, Pierre-Yves Vandenbussche, and Bernard Vatant. Enabling linked-data publication with the datalift platform. In *26th Conference on Artificial Intelligence (AAAI-12)*, 2012. 50, 122, 128, 142

[38] Pierre-Yves Vandenbussche, Bernard Vatant, and L. Rozat. Linked open vocabularies: an initiative for the web of data. In *QetR Workshop*, Chambery, France, 2011. 52, 85

[39] Bert Van Nuffelen, Valentina Janev, Michael Martin, Vuk Mijovic, and Sebastian Tramp. Supporting the linked data life cycle using an integrated tool stack. In SÃ¶ren Auer, Volha Bryl, and Sebastian Tramp, editors, *Linked Open Data – Creating Knowledge Out of Interlinked Data*, Lecture Notes in Computer Science, pages 108–129. Springer International Publishing, 2014. 55

[40] Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes - a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011. 56

[41] Tommaso Soru and Axel-Cyrille Ngonga Ngomo. Rapid execution of weighted edit distances. In *Proceedings of the Ontology Matching Workshop*, 2013. 56

[42] Axel-Cyrille Ngonga Ngomo. Orchid - reduction-ratio-optimal computation of geospatial distances for link discovery. In *Proceedings of ISWC 2013*, 2013. 56, 58

[43] GeoKnow Project. Spatial mapping framework for enriching rdf datasets with geospatial information. Manual-Deliverable, 2014. `https://github.com/GeoKnow/GeoLift/blob/master/GeoLiftManual/GeoLiftManual.pdf`. 56

[44] A. Jentzsch, R. Isele, and C. Bizer. Silk-generating rdf links while publishing or consuming linked data. In *Poster at the International Semantic Web Conference (ISWC2010), Shanghai*, 2010. 58, 108

[45] EU ISA Programme Core Vocabularies Working Group (Location Task Force). Isa programme location vocabulary. W3C document, 2013. `http://www.w3.org/ns/locn`. 60

[46] Marti A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009. 69

[47] Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearinged, and Ka-Ping Yee. Finding the flow in web site search, 2002. 69

[48] Lars Grammel and Margaret-Anne Storey. Choosel â web-based visualization construction and coordination for information visualization novices. Poster: IEEE Information Visualization Conference, 2010. 70

[49] IBM Research. Many eyes, 2010. `http://www-958.ibm.com/software/data/cognos/manyeyes/`. 70

[50] Mike Bostock. Data-driven documents, 2012. `http://d3js.org/`. 70

[51] Tetherless Constellation. How to use google visualization api, 2012. http://data-gov.tw.rpi.edu/wiki/$\mathrm{How}_t o_u se_G oogle_V isualization_A PI$.71

[52] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011. 72

[53] G. Martin Skjaeveland. Sgvizler: A javascript wrapper for easy visualization of sparql result sets. In *9^{th} Extended Semantic Web Conference (ESWC'12)*, 2012. 72, 91

[54] Claus Stadler, Michael Martin, and Sören Auer. Exploring the Web of Spatial Data with Facete. In *Companion proceedings of 23rd International World Wide Web Conference (WWW)*, pages 175–178, 2014. 73, 75

[55] Alvaro Graves. Creation of visualizations based on linked data. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, WIMS '13, pages 41:1–41:12, New York, NY, USA, 2013. ACM. 73

[56] Jakub Klímek, Jirí Helmich, and Martin Necaský. Payola: Collaborative linked data analysis and visualization framework. In *The Semantic Web: ESWC 2013 Satellite Events - ESWC 2013 Satellite Events, Montpellier, France, May 26-30, 2013, Revised Selected Papers*, pages 147–151, 2013. 74

[57] Josep Maria Brunetti, Sören Auer, Roberto García, Jakub Klímek, and Martin Nečaský. Formal linked data visualization model. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, IIWAS '13, pages 309:309–309:318, New York, NY, USA, 2013. ACM. 74

[58] Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 69–, Washington, DC, USA, 2000. IEEE Computer Society. 74

[59] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web Journal*, 2(2):89–124, 2011. 74

[60] J. Klimek, J. Helmich, and M. Neasky. Application of the Linked Data Visualization Model on Real World Data from the Czech LOD Cloud. In *6^{th} International Workshop on the Linked Data on the Web (LDOW'14)*, 2014. 74

[61] Percy E Salas, Michael Martin, Fernando Maia Da Mota, Karin Breitman, Sören Auer, and Marco A Casanova. Publishing statistical data on the web. In *Proceedings of 6th International IEEE Conference on Semantic Computing*, IEEE 2012. IEEE, 2012. 74

[62] Alexander de Leon, Filip Wisniewki, Boris Villazón-Terrazas, and Oscar Corcho. Map4rdf - Faceted Browser for Geospatial Datasets. In *Using Open Data: policy modeling, citizen empowerment, data journalism (PMOD'12)*, 2012. 74

[63] Danh Le Phuoc, Axel Polleres, Christian Morbidoni, Manfred Hauswirth, and Giovanni Tummarello. Rapid semantic web mashup development through semantic web pipes. In *18th International World Wide Web Conference (WWW'09)*, Madrid, Spain, 2009. 75

[64] G. Atemezing and R. Troncy. Tools for visualization (v.1.2). Deliverables - D.6.2 of DataLift project, 2012. 76

[65] Fadi Maali and John Erickson (eds). Data catalog vocabulary (dcat). W3C Recommendation, 2014. `http://www.w3.org/TR/vocab-dcat/`. 9, 79

[66] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. 80

[67] Emmanuel Pietriga. Isaviz: A visual authoring tool for rdf, 2004. `http://www.w3.org/2001/11/IsaViz/`. 80

[68] Tim Berners-lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *In Proceedings of the 3rd International Semantic Web User Interaction Workshop*, 2006. 80

[69] G. Atemezing and R. Troncy. Usage scenarii for applications (v.1.1). Deliverables-D.6.1 of DataLift project, 2012. 81

[70] Colin Ware. *Information Visualization, Second Edition: Perception for Design*. Morgan Kaufmann Publishers Inc.; 2 edition (April 21, 2004), San Francisco, CA, USA, 2014. 83

[71] B Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Symposium on Visual Languages '96, IEEE, Los Alamos, CA (September 1996)*, pages 336–343, 1996. 83, 85

[72] Jeni Tennison. Guest post: A developers' guide to the linked data apis, July 2010. `http://data.gov.uk/blog/guest-post-developers-guide-linked-data-apis-jeni-tennison`. 83

[73] Pierre-Yves Vandenbussche and Bernard Vatant. Metadata Recommendations For Linked Open Vocabularies. OKFN, 2012. `http://lov.okfn.org/dataset/lov/Recommendations_Vocabulary_Design.pdf`. 88

[74] Carlos Buil-Aranda, Aidan Hogan, Jürgen Umbrich, and Pierre-Yves Vandenbussche. Sparql web-querying infrastructure: Ready for action? In *The Semantic Web–ISWC 2013*, pages 277–293. Springer, 2013. 90

[75] Emmanuel Pietriga, Christian Bizer, David Karger, and Ryan Lee. Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In *5th International Semantic Web Conference (ISWC'06)*, pages 158–171, 2006. 96

[76] Mike Bergman. Deconstructing the Google Knowledge Graph. `http://www.mkbergman.com/1009/deconstructing-the-google-knowledge-graph`. 96

[77] Raphaël Troncy, Bartosz Malocha, and André Fialho. Linking Events with Media. In *6th International Conference on Semantic Systems (I-SEMANTICS'10)*, Graz, Austria, 2010. 105

[78] Pieter Colpert, Anastasia Dimou, Ghislain Auguste Atemezing, and Raphaël Troncy. Technical toolset for open data competitions apps for europe. Technical report, 2014. `http://www.appsforeurope.eu/sites/default/files/TechnicaltoolsetforopendatacompetitionsAppsforEurope_0.pdf`. 111

[79] Bernadette Hyland and David Wood. The joy of data - cookbook for publishing linked government data on the web, 2011. `http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook`. 121

[80] Michael Hausenblas and Richard Cygankiak. Linked data life cycles, 2012. `http://linked-data-life-cycles.info/`. 121

[81] Boris Villazón-Terrazas; et al. Methodological guidelines for publishing government linked data. `http://link.springer.com/chapter/10.1007/978-1-4614-1767-5_2`. 122

[82] Bernadette Hyland, Ghislain Atemezing, Michael Pendleton, and Biplav Srivastava (eds). Linked data glossary. W3C Working Group Note, 2013. `http://www.w3.org/TR/ld-glossary/`. 9, 123

[83] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5:1–22, 2009. 128, 142

[84] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets. In *2nd Workshop on Linked Data on the Web (LDOW)*, Madrid, Spain, 2009. 128

[85] Bernard Vatant and Pierre-Yves Vandenbussche. Catalogue de Vocabulaires. Datalift, D2.2, 2013. `http://datalift.org/en/node/18`. 134

[86] Eric Prud'hommeaux and Carlos Buil-Aranda. SPARQL 1.1 Federated Query. W3C Recommendation, 2013. `http://www.w3.org/TR/sparql11-federated-query/`. 135

[87] S. M. Ross. A First Course in Probability, 2002. 143

[88] R. Meymandpour and J. G. Davis. Ranking Universities Using Linked Open Data. In *5th International Workshop on the Linked Data on the Web (LDOW'13)*, 2013. 143

[89] Johann Schaible, Thomas Gottron, and Ansgar Scherp. Survey on Common Strategies of Vocabulary Reuse in Linked Open Data Modeling. In *11th Extended Semantic Web Conference (ESWC'14)*, pages 457–472, 2014. 149

[90] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011. 150

[91] Guido Governatori, Antonino Rotolo, Serena Villata, and Fabien Gandon. One license to compose them all - a deontic logic approach to data licensing on the web of data. In *International Semantic Web Conference (1)*, volume 8218 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2013. 151, 152, 153, 155, 157

[92] Víctor Rodríguez-Doncel, Asunción Gómez-Pérez, and Nandana Mihindukulasooriya. Rights declaration in linked data. In Hartig et al. [105]. 152, 153

[93] G. R. Gangadharan, V. D'Andrea, R. Iannella, and M. Weiss. Odrl service licensing profile (ODRL-S). In *Proceedings of Virtual Goods*, 2007. 153

[94] G. R. Gangadharan, Michael Weiss, Vincenzo D'Andrea, and Renato Iannella. Service license composition and compatibility analysis. In *Proceedings of ICSOC, LNCS 4749*, pages 257–269. Springer, 2007. 153

[95] Nadia Nadah, Mélanie Dulong de Rosnay, and Bruno Bachimont. Licensing digital content with a generic ontology: escaping from the jungle of rights expression languages. In *Proceedings of ICAIL* [106], pages 65–69. 153

[96] Hong Linh Truong, G. R. Gangadharan, Marco Comerio, Schahram Dustdar, and Flavio De Paoli. On analyzing and developing data contracts in cloud-based data marketplaces. In *Proceedings of APSCC, IEEE*, pages 174–181, 2011. 153

[97] Markus Krötzsch and Sebastian Speiser. ShareAlike Your Data: Self-referential Usage Policies for the Semantic Web. In *Proceedings of ISWC, LNCS 7031*, pages 354–369. Springer, 2011. 153

[98] Thomas F. Gordon. Analyzing open source license compatibility issues with Carneades. In *Proceedings of ICAIL*, pages 51–55. ACM, 2011. 153

[99] V. Rodriguez-Doncel, M.C. Suarez Figueroa, A. Gomez-Perez, and M. Poveda Villalon. Licensing patterns for linked data. In *Proc. of the 4th International Workshop on Ontology Patterns*, 2013. 153

[100] V. Rodriguez-Doncel, M.C. Suarez Figueroa, A. Gomez-Perez, and M. Poveda Villalon. License linked data resources pattern. In *Proc. of the 4th International Workshop on Ontology Patterns*, 2013. 153

[101] Riccardo Pucella and Vicky Weissman. A logic for reasoning about digital rights. In *Proceedings of CSFW*, pages 282–294. IEEE, 2002. 153

[102] Elena Cabrio, Alessio Palmero Aprosio, and Serena Villata. These are your rights: A natural language processing approach to automated rdf licenses generation. In *ESWC2014, LNCS*, 2014. 154, 155

[103] Ho-Pun Lam and Guido Governatori. The making of SPINdle. In *Proceedings of RuleML, LNCS 5858*, pages 315–322. Springer, 2009. 155

[104] Harith Alani, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors. *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, volume 8219 of *Lecture Notes in Computer Science*. Springer, 2013. 169

[105] Olaf Hartig, Juan Sequeda, Aidan Hogan, and Takahide Matsutsuka, editors. *Proceedings of the Fourth International Workshop on Consuming Linked Data, COLD 2013, Sydney, Australia, October 22, 2013*, volume 1034 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013. 174

[106] *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA*. ACM, 2007. 174

[107] Jan Van den Bussche and Victor Vianu, editors. *Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings*, volume 1973 of *Lecture Notes in Computer Science*. Springer, 2001. 179

[108] Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel F. Cruz, editors. *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010*, volume 689 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010. 180

[109] Matteo Baldoni, Jamal Bentahar, M. Birna van Riemsdijk, and John Lloyd, editors. *Declarative Agent Languages and Technologies VII, 7th International Workshop, DALT 2009, Budapest, Hungary, May 11, 2009. Revised Selected and Invited Papers*, volume 5948 of *Lecture Notes in Computer Science*. Springer, 2010. 180

[110] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors. *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009. 180

[111] *9th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2008), 2-4 June 2008, Palisades, New York, USA*. IEEE Computer Society, 2008. 181

[112] Sadok Ben Yahia and Jean-Marc Petit, editors. *Extraction et gestion des connaissances (EGC'2010), Actes, 26 au 29 janvier 2010, Hammamet, Tunisie*, volume RNTI-E-19 of *Revue des Nouvelles Technologies de l'Information*. Cépaduès-Éditions, 2010. 181

[113] Birte Glimm and David Huynh, editors. *Proceedings of the ISWC 2012 Posters & Demonstrations Track, Boston, USA, November 11-15, 2012*, volume 914 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012. 182

[114] Antonino Rotolo, Serena Villata, and Fabien Gandon. A deontic logic semantics for licenses composition in the web of data. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, pages 111–120. ACM, 2013.

[115] P.Y. Vandenbussche, B. Vatant, and L. Rozat. Linked open vocabularies: an initiative for the web of data. In *QetR Workshop, Chambery, France.*, 2011.

[116] Guido Governatori, Antonino Rotolo, Serena Villata, and Fabien G. One license to compose them all a deontic logic approach to data licensing on the web of data, 2012.

[117] Pierre-Yves Vandenbussche and Bernard Vatant. Metadata Recommendations For Linked Open Vocabularies. OKFN, 2012. `http://lov.okfn.org/dataset/lov/Recommendations_Vocabulary_Design.pdf`.

[118] Krzysztof Janowicz, Sven Schade, Arne Bröring, Carsten Kessler, Christoph Stasch, Patrick Maué, and Thorsten Diekhof. A transparent semantic enablement layer for the geospatial web. In *2nd International Terra Cognita Workshop*, 2009.

[119] International Organization for Standardization (TC 211). ISO 19107: Geographic information - Spatial Schema., 2003.

[120] International Organization for Standardization (TC 211). ISO 19109: Geographic information - Rules for application schema., 2005.

[121] International Organization for Standardization (TC 211). ISO 19111: Geographic information - Spatial referencing by coordinates, 2007.

[122] Bénédicte Bucher, Nathalie Abadie, and Ghislain Auguste Atemezing. Modélisation de connaissances spécifiques: information spatiale. Rapport de Recherche. Délivrable WP2.T.2.3- Projet Datalift, 2013.

[123] W3C Semantic Web Interest Group (SWIG). Wgs84 rdf geoposition vocabulary, 2004. `http://www.w3.org/2003/01/geo/`.

[124] Tim Berners-lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *In Proceedings of the 3rd International Semantic Web User Interaction Workshop*, 2006.

[125] S. Harris and A Seaborne. SPARQL 1.1 Query Language, 2013. `http://www.w3.org/TR/sparql11-query/`.

[126] Vladimir Geroimenko and Chaomei Chen. Visualising the semantic web, 2006.

[127] Josep Maria Brunetti, Soren Auer, and Roberto Garcia. The linked data visualization model. In *In Proceedings of the 11th International Semantic Web Conference*, 2012.

[128] Antoine Isaac, William Waites, Jeff Young, and Marcia Zeng. Library Linked Data Incubator Group: Datasets, Value Vocabularies, and Metadata Element Sets. W3C Incubator Group Report, 2011. `http://www.w3.org/2005/Incubator/lld/XGR-lld-vocabdataset/`.

[129] Sean Bechhofer and Alistair Mile. SKOS Simple Knowledge Organization System Reference. W3C Recommendation, 2009. `http://www.w3.org/TR/skos-reference/`.

[130] Bernadette Hyland, Ghislain Auguste Atemezing, Michael Pendleton, and Biplav Srivastava. Linked Data Glossary. W3C Working Group Note, 2013. `http://www.w3.org/TR/ld-glossary/`.

[131] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space: Theory and Technology.* Morgan & Claypool Publishers, 2011.

[132] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and Maintaining Links on the Web of Data. In *8^{th} International Semantic Web Conference (ISWC)*, Washington DC, USA, 2009.

[133] Thomas Steiner and Stefan Mirea. SEKI@home or Crowdsourcing an Open Knowledge Graph. In *1^{st} International Workshop on Knowledge Extraction & Consolidation from Social Media (KECSM'12)*, Boston, USA, 2012.

[134] Ashutosh Jadhav, Hemant Purohit, Pramod Ananthram, Ajith Ranabahu, Vinh Nguyen, Pablo Mendes, Alan Gary Smith, Michael Cooney, and Amit Sheth. Twitris 2.0 : Semantically Empowered System for Understanding Perceptions From Social Data. In *Semantic Web Challenge at the 9^{th} International Semantic Web Conference (ISWC'10)*, Shanghai, China, 2010.

[135] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *3rd ACM International Conference on Web Search and Data Mining*, pages 291–300, New York, NY, USA, 2010.

[136] Smitashree Choudhury and John G. Breslin. Extracting Semantic Entities and Events from Sports Tweets. In *Making Sense of Microposts (#MSM2011)*, 2011.

[137] Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: a brief history. In *16th International Conference on Computational linguistics (COLING'96)*, pages 466–471, Copenhagen, Denmark, 1996.

[138] Xueliang Liu, Raphaël Troncy, and Benoît Huet. Using Social Media to Identify Events. In *3rd Workshop on Social Media (WSM'11)*, Scottsdale, Arizona, USA, 2011.

[139] Pablo Mendes, Alexandre Passant, and Pavan Kapanipathi. TWARQL: Tapping into the Wisdom of the Crowd. In *6th International Conference on Semantic Systems (I-SEMANTICS'10)*, Graz, Austria, 2010.

[140] Knud Möller, Tom Heath, Siegfried Handschuh, and John Domingue. Recipes for Semantic Web dog food - The ESWC and ISWC metadata projects. In *6th International Semantic Web Conference (ISWC'07)*, pages 802–815, Busan, Korea, 2007.

[141] Giuseppe Rizzo and Raphaël Troncy. NERD: A Framework for Evaluating Named Entity Recognition Tools in the Web of Data. In *10th International Semantic Web Conference (ISWC'11), Demo Session*, pages 1–4, Bonn, Germany, 2011.

[142] Matthew Rowe and Milan Stankovic. Aligning Tweets with Events: Automation via Semantics. *Semantic Web Journal*, 2011.

[143] R. Shaw, R. Troncy, and L. Hardman. LODE: Linking Open Descriptions of Events. In *4th Asian Semantic Web Conference (ASWC'09)*, pages 153–167, Shanghai, China, 2009.

[144] Milan Stankovic. Modeling Online Presence. In *1st Social Data on the Web Workshop (SDoW'08)*, 2008.

[145] Katrin Weller, Evelyn Drge, and Cornelius Puschmann. Citation Analysis in Twitter: Approaches for Defining and Measuring Information Flows within Tweets during Scientific Conferences. In *Making Sense of Microposts (#MSM2011)*, pages 1–12, 2011.

[146] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, pages 1524–1534, 2011.

[147] Guido Governatori, Ho-Pun Lam, Antonino Rotolo, Serena Villata, and Fabien Gandon. Heuristics for licenses composition. In *Proceedings of JURIX*, pages 77–86. IOS Press, 2013.

[148] Kevin D. Ashley, editor. *Legal Knowledge and Information Systems - JURIX 2013: The Twenty-Sixth Annual Conference, December 11-13, 2013, University of Bologna, Italy*, volume 259 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2013.

[149] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2010.

[150] Paul T. Groth, Yolanda Gil, James Cheney, and Simon Miles. Requirements for provenance on the web. *IJDC*, 7(1):39–56, 2012.

[151] Michael J. Maher, Andrew Rock, Grigoris Antoniou, David Billington, and Tristan Miller. Efficient defeasible reasoning systems. *International Journal of Artificial Intelligence Tools*, 10:483–501, 2001.

[152] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and where: A characterization of data provenance. In den Bussche and Vianu [107], pages 316–330.

[153] Marco Comerio. *Web Service Contracts: Specification, Selection and Composition*. PhD thesis, University of Milano-Bicocca, 2009.

[154] Guido Governatori. On the relationship between carneades and defeasible logic. In *Proceedings of ICAIL*, pages 31–40. ACM, 2011.

[155] Kevin D. Ashley and Tom M. van Engers, editors. *The 13th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 6-10, 2011, Pittsburgh, PA, USA*. ACM, 2011.

[156] G. R. Gangadharan, Hong Linh Truong, Martin Treiber, Vincenzo D'Andrea, Schahram Dustdar, Renato Iannella, and Michael Weiss. Consumer-specified service license selection and composition. In *Proceedings of ICCBSS, IEEE*, pages 194–203, 2008.

[157] *Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008), February, 25-29, 2008, Madrid, Spain, Proceedings*. IEEE Computer Society, 2008.

[158] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[159] James J. Park, Christos Nikolaou, and Jiannong Cao, editors. *2011 IEEE Asia-Pacific Services Computing Conference, APSCC 2011, Jeju, Korea (South), December 12-15, 2011*. IEEE, 2011.

[160] Bernd J. Krämer, Kwei-Jay Lin, and Priya Narasimhan, editors. *Service-Oriented Computing - ICSOC 2007, Fifth International Conference, Vienna, Austria, September 17-20, 2007, Proceedings*, volume 4749 of *Lecture Notes in Computer Science*. Springer, 2007.

[161] Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors. *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, volume 7031 of *Lecture Notes in Computer Science*. Springer, 2011.

[162] Cássia Trojahn dos Santos and Jérôme Euzenat. Consistency-driven argumentation for alignment agreement. In Shvaiko et al. [108].

[163] Nicoletta Fornara and Marco Colombetti. Ontology and time evolution of obligations and prohibitions using semantic web technology. In Baldoni et al. [109], pages 101–118.

[164] Moreau et al. *The Open Provenance Model Core Specification (v1.1)*, 2009. http://eprints.ecs.soton.ac.uk/18332/1/opm.pdf.

[165] Jun Zhao. *Guide to the Open Provenance Vocabulary*, 2010. http://purl.org/net/opmv/guide.

[166] P. Miller, R. Styles, and T. Heath. Open data commons, a license for open data. In *LDOW*, 2008.

[167] Renato Iannella. *Open Digital Rights Language (ODRL)*, 2002. http://odrl.net/1.1/ODRL-11.pdf.

[168] Hal Abelson, Ben Adida, Mike Linksvayer, and Nathan Yergler. *ccREL: The Creative Commons Rights Expression Language*, 2008.

[169] *ODC Public Domain Dedication and License*, 2008. http://download.opencontentlawyer.com/ODC_PDDL.pdf.

[170] Rui Zhang, Alessandro Artale, Fausto Giunchiglia, and Bruno Crispo. Using description logics in relation based access control. In Grau et al. [110].

[171] Owen Sacco and Alexandre Passant. A Privacy Preference Ontology (PPO) for Linked Data. In *Proceedings of the 4th Workshop about Linked Data on the Web (LDOW-2011)*, 2011.

[172] James Hollenbach, Joe Presbrey, and Tim Berners-Lee. Using RDF Metadata To Enable Access Control on the Social Semantic Web. In *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK-2009)*, 2009.

[173] Fabian Abel, Juri Luca De Coi, Nicola Henze, Arne Wolf Koesling, Daniel Krause, and Daniel Olmedilla. Enabling advanced and context-dependent access control in rdf stores. In *Proceedings of the 6th International Semantic Web Conference (ISWC-2007), LNCS 4825*, pages 1–14, 2007.

[174] Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors. *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*. Springer, 2007.

[175] Hannes Muhleisen, Martin Kost, and Johann-Christoph Freytag. SWRL-based Access Policies for Linked Data. In *Proceedings of the 2nd Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2010)*, 2010.

[176] Stephanie Stroka, Sebastian Schaffert, and Tobias Burger. Access Control in the Social Semantic Web - Extending the idea of FOAF+SSL in KiWi. In *Proceedings of the 2nd Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2010)*, 2010.

[177] Christian Bizer, Tom Heath, and Tim Berners-lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5:1–22.

[178] Juri Luca De Coi, Daniel Olmedilla, Sergej Zerr, Piero A. Bonatti, and Luigi Sauro. A trust management package for policy-driven protection & personalization of web content. In *POLICY* [111], pages 228–230.

[179] Olaf Hartig. Querying trust in rdf data with tsparql. In *Proceedings of the 6th European Semantic Web Conference (ESWC-2009), LNCS 5554*, pages 5–20, 2009.

[180] Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors. *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*, volume 5554 of *Lecture Notes in Computer Science*. Springer, 2009.

[181] Jeremy Carroll, Christian Bizer, Patrick Hayes, and Patrick Stickler. Named graphs. *J. Web Sem.*, 3(4):247–267, 2005.

[182] Michel Buffa, Catherine Faron-Zucker, and Anna Kolomoyskaya. Gestion sémantique des droits d'accès au contenu : l'ontologie AMO. In Yahia and Petit [112], pages 471–482.

[183] Fausto Giunchiglia, Rui Zhang, and Bruno Crispo. Ontology driven community access control. In *Proceedings of the 1st Workshop on Trust and Privacy on the Social and Semantic Web (SPOT-2009)*, 2009.

[184] John Breslin, Alexandre Passant, and Stefan Decker. *The Social Semantic Web*. Springer, Heidelberg, 2009.

[185] Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, and Bhavani M. Thuraisingham. Semantic web-based social network access control. *Computers & Security*, 30(2-3):108–115, 2011.

[186] Paul Miller, Rob Styles, and Tom Heath. Open data commons, a license for open data. In *Proceedings of LDOW*, 2008.

[187] Hal Abelson, Ben Adida, Mike Linksvayer, and Nathen Yergler. ccREL: The creative commons rights expression language. Technical report, 2008.

[188] ODC Public Domain Dedication and License. Technical report, 2008.

[189] Raul Palma, Jens Hartmann, and Peter Haase. OMV Ontology Metadata Vocabulary for the Semantic Web. Technical report, 2008.

[190] Richard Raysman, Edward A. Pisacreta, and Kenneth A. Adler. *Intellectual Property Licensing: Forms and Analysis*. Law Journal Press, 1999.

[191] Serena Villata and Fabien Gandon. Towards licenses compatibility and composition in the web of data. In Glimm and Huynh [113].

[192] Serena Villata and Fabien Gandon. Licenses compatibility and composition in the web of data. In *Proceedings of COLD*, CEUR Workshop Proceedings 905, 2012.

[193] Juan Sequeda, Andreas Harth, and Olaf Hartig, editors. *Proceedings of the Third International Workshop on Consuming Linked Data, COLD 2012, Boston, MA, USA, November 12, 2012*, volume 905 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[194] Harith Alani, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors. *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*. Springer, 2013.

[195] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing with compositional vector grammars. In *ACL (1)*, pages 455–465, 2013.

[196] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Extracting multilingual natural-language patterns for rdf predicates. In *EKAW*, pages 87–96, 2012.