Terra Cognita 2012 Workshop

In Conjunction with the 11th International Semantic Web Conference (ISWC 2012)

Workshop Proceedings

Boston, USA

November 12, 2012

Introduction

The wide availability of technologies such as GPS, map services and social networks, has resulted in the proliferation of geospatial data on the Web. In addition to material produced by professionals (e.g., maps), the public has also been encouraged to make geospatial content, including their geographical location, available online. The volume of such user-generated geospatial content is constantly growing. Similarly, the amount of data extracted from the Web and published as Linked Open Data is increasing. Linked Open Data include many data sets with geospatial properties such as coordinates, feature classes or topological relations. Examples of such data sets are GeoNames.org, LinkedGeoData.org and DBpedia.org.

The geo-referencing of Web resources and users has given rise to various services and applications that exploit it. With the location of users being made available widely, new issues such as those pertaining to security and privacy arise. Likewise, emergency response, context sensitive user applications, and complex GIS tasks all lend themselves toward solutions that combine both the Geospatial Web and the Semantic Web.

Researchers have been quick to realize the importance of these developments and have started working on the relevant research problems, giving rise to new topical research areas such as Geographic Information Retrieval, Linked Geospatial Data, GeoWeb 2.0. Similarly, standardization bodies such as the Open Geospatial Consortium (OGC) have been developing relevant standards such as the Geography Markup Language (GML) and GeoSPARQL.

The workshop will bring together researchers and practitioners from various disciplines, as well as interested parties from industry and government, to advance the frontiers of this exciting research area. Bringing together Semantic Web and geospatial researchers helps encourage the use of semantics in geospatial applications and the use of spatial elements in semantic research and applications. The field continues to gain popularity, resulting in a need for a forum to discuss relevant issues.

Topics Of Interest

Topics of interest include, but are not limited to:

- Data models and languages for the Geospatial Web
- Systems and architectures for the Geospatial Web
- Geographic Information Retrieval
- Linked Geospatial Data
- Ontologies and rules in the Geospatial Web
- Uncertainty in the Geospatial Web
- User interface technologies for the Geospatial Web
- Geospatial Web and mobile data management
- Security and privacy issues in the Geospatial Web
- Geospatial Web applications
- User-generated geospatial content
- OGC and W3C technologies and standards in the Geospatial Web

Workshop Chairs

- Dave Kolas, BBN Technologies, U.S.A
- Matthew Perry, Oracle Corp., Nashua, NH, U.S.A.
- Rolf Grütter, Swiss Federal Research Institute WSL, Birmensdorf, Switzerland
- Manolis Koubarakis National and Kapodistrian University of Athens, Greece

Program Committee

Alia Abdelmoty, Cardiff University, UK
Thomas Barkowsky, University of Bremen, Germany
Oscar Corcho, Universidad Politécnica de Madrid, Spain
Isabel Cruz, University of Illinois, Chicago
Mike Dean, BBN Technologies, USA
John Goodwin, Ordnance Survey, UK
Glen Hart, Ordnance Survey, UK
Krzysztof Janowicz, University of California, Santa Barbara, USA
Marinos Kavouras, National Technical University of Athens, Greece
Stefan Manegold, CWI, The Netherlands
Alexandros Ntoulas, Microsoft Research
Dieter Pfoser, Athena-Research and Innovation Center, Greece
Florian Probst, SAP Research, Germany
Ross Purves, University of Zurich, Switzerland
Özguer L. Özcep, Hamburg University of Technology, Germany
Thorsten Reitz, Fraunhofer Institute for Computer Graphics, Germany
Timos Sellis, Athena-Research and Innovation Center and National Technical University of Athens,Greece.
Spiros Skiadopoulos, University of the Peloponnese, Greece
Stavros Vassos, National and Kapodistrian University of Athens, Greece
Nancy Wiegand, University of Wisconsin, USA
James Wilson, James Madison University, USA
Stefan Woelfl, University of Freiburg, Germany

Organi zation/Sponsorship

This workshop is organized by members of the Spatial Ontology Community of Practice (SOCoP) and European project TELEIOS.

TELEIOS is an FP7/ICT project with the goal of building an Earth Observatory. TELEIOS concentrates heavily on geospatial data (sattelite images, traditional GIS data, geospatial Web data).

SOCoP is a geospatial semantics interest group currently mainly with members from U.S. federal agencies, academia, and business. SOCoP's goal is to foster collaboration among users, technologists and researchers of spatial knowledge representations and reasoning towards the development of a set of core, common geospatial ontologies for use by all in the Semantic Web.

Program

9:00 – 10:00

The GeoSPARQL OGC standard

Matthew Perry (Oracle)

10:00 – 10:30

The Parliament implementation of GeoSPARQL

Dave Kolas (Raytheon)

10:30 – 11:00

Break: Foyer

11:00 – 12:30

Ghislain Auguste Atemezing and Raphaël Troncy
Comparing Vocabularies for Representing Geographical Features and Their Geometry

Charalampos Nikolaou and Manolis Koubarakis.
Querying Linked Geospatial Data with Incomplete Information

Jesper Zedlitz and Norbert Luttenberger.
Transforming Between UML Conceptual Models And OWL 2 Ontologies

12:30 – 14:00

Lunch: Foyer

14:00 – 15:00

Spatial Description Logics (Tentative)

Ralf Moeller (Hamburg University of Technology)

15:00 – 15:30

Marek Šmíd and Zdeněk Kouba
OnGIS: Ontology Driven Geospatial Search and Integration

15:30 – 16:00

Break: Foyer

16:00 – 17:30

Alkyoni Baglatzi, Margarita Kokla and Marinos Kavouras
Semantifying OpenStreetMap

Mihai Codescu, Daniel Couto Vale, Oliver Kutz and Till Mossakowski
Ontology-based Route Planning for OpenStreetMap

Kostis Kyzirakos
Invited Demo: The Strabon Implementation of GeoSPARQL

18:00

Closing Session

Table of Contents

Comparing Vocabularies for Representing Geographical Features and Their Geometry

Ghislain Auguste Atemezing, Raphaël Troncy

EURECOM, Sophia Antipolis, France,
{auguste.atemezing, raphael.troncy}@eurecom.fr

Abstract. 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 this paper, 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 in 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. We 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 conclude with some challenges to be taken into account when dealing with the descriptions of complex geometries.

Keywords: Geodata, GeoSPARQL, Geographic information, Schema Alignment, Datalift

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 descriptions of addresses, buildings, topographic information, occupied zones, etc. A few years ago, IGN has developed a core ontology named GeOnto for describing all types of buildings located in the French territory. Integrating these databases 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 paper, we address some of these uses cases, starting from the selection of the right vocabularies to represent the data and their alignment to ease future dataset interlinking. We first analyze the use of geographical information in the web of data (Section 2). Then, we survey the existing approaches for modeling both the features and their geometries (Section 3). We define the scenario of modeling the 7^{th} arrondissement of Paris to highlight the diversity of these approaches (Section 4). We then propose alignments between vocabularies to describe features or points of interest using GeOnto as our pivot ontology (Section 5). To address geometry modeling, we also survey existing approaches, leading to an extension of GeOnto to support geometry. We look at the triple stores supporting all types of geometry and discuss some challenging issues regarding geodata as the GeoSPARQL[1] standard has recently been adopted by the Open Geospatial Consortium (Section 6). Finally, we give our conclusions and outline future work (Section 7).

2 Geographic information in the Web of Data

2.1 LOD Cloud Review

The recent publication of statistics concerning the actual usage of vocabularies on the LOD cloud[2] provides not only an overview of best practice usage recommended by Tim Berners-Lee[3], 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[4] is the most widely used vocabulary, followed by the `spatialrelations`[5] 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

[1] `http://www.opengeospatial.org/standards/geosparql`
[2] `http://stats.lod2.eu`
[3] `http://www.w3.org/DesignIssues/LinkedData.html`
[4] `http://www.w3.org/2003/01/geo/wgs84_pos`
[5] `http://data.ordnancesurvey.co.uk/ontology/spatialrelations`

`rdfs:label`(1, 586, 115, 316 triples). This shows the importance of geodata on the web. Table 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 using	#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 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.

2.2 Geodata Provider 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 2 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 project data	682 605 triples	Web Service
NUTS project data	316 238 triples	Web Service
IGN experimental	629 716 triples	SPARQL endpoint

Table 2. Geodata by provider and their different access type

3 Geodata Modeling Approach

3.1 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 used in the Geonames ontology[6] 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[7] 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` and `Waterway`. The same approach is used for the French GeOnto ontology (Section 5), 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[8], for statistics decomposition[9] and for postal codes[10]. 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[11] or Google Place categories[12]. For this last approach, we have built an associated OWL vocabulary composed of alignments with other vocabularies.

3.2 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 [8]:

- *Point representation*: the classical way to represent a location by providing the latitude and longitude in a given coordinate reference system (the most

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

used on the web is the WGS84 datum represented in RDF by the W3C 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[13] 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 [1].
- *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 [3].
- *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 [4].
- *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[14].

4 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 government ministries[15]. 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[16] geodetic system.

4.1 DBpedia Modeling

We provide below an excerpt of the DBpedia description for this resource.

[13] `http://www.fao.org/countryprofiles/geoinfo/geopolitical/resource/`
[14] `http://geovocab.org/doc/neogeo/`
[15] `http://en.wikipedia.org/wiki/7th_arrondissement_of_Paris`
[16] `http://en.wikipedia.org/wiki/World_Geodetic_System`

```
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:département  dbpedia:Paris ;
  dbpprop:région dbpedia:Île-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 7^{th} arrondissement is a simple POINT defined by a latitude and a longitude.

4.2 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 "7ème 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";
  geo:lat "48.8565";
  geo:long "2.321".
```

4.3 LinkedGeoData Modeling

In LinkedGeoData, the district is a `lgdo:Suburb rdfs:subClassOf 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 .
```

4.4 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 7^{th} 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.

5 Aligning Geo Vocabularies

IGN is a public service in France in charge of describing, from the physical and geometry point of view, the surface of the French territory and the occupation of the land, and to elaborate and update continuously the forestal resources. They are also experimenting in exposing some of their data as Linked Data and act as an important provider in the `http://data.gouv.fr` portal.

5.1 Existing Vocabularies

IGN has developed two complementary vocabularies (GeOnto and bdtopo) which differ in their provenance but have the same scope, which is to describe geographic entities in the French territory. GeOnto is the product of a research project[17] aiming at building and aligning heterogeneous ontologies in the geographic domain. The "light" version of the final ontology[18] defines two top classes for a total of 783 classes and 17 properties (12 DP / 5 OP). GeOnto has labels in both French and English, but has no comments specified for the resources. The bdtopo ontology is derived from a geospatial database with the same name. It contains 237 classes and 51 properties (47 DP / 4 OP). All the labels and comments are in French.

5.2 GeOnto Alignment Process

The first step towards interoperability of French geographic features and the existing vocabularies is to align GeOnto to other vocabularies. We choose GeOnto because it covers a large number of categories and also has labels in English. We have performed the alignment with five OWL vocabularies (bdtopo, LGD, DBpedia, Schema.org and Geonames) and two flat taxonomies (Foursquare, Google Place). For the latter, we have transformed the flat list of types and categories into an OWL ontology. For each alignment performed, we only consider `owl:equivalentClass` axioms. We use the Silk tool [9] to compute the alignment using two metrics for string comparison: the *levenshteinDistance* and *jaro* distances. They work on the English labels except for the alignment with bdtopo where we use the French labels. We apply the average aggregation function on these metrics with an empirically derived threshold. However, for generating

[17] `http://geonto.lri.fr/Livrables.html`
[18] `http://semantics.eurecom.fr/datalift/tc2012/vocabs/GeoOnto/`

the final mapping file for vocabularies of small size, we manually validate and insert relations of type `rdfs:subClassOf`. The threshold to validate the results is set to 100% for links considered to be correct and greater than 40% for links to be verified. The alignment with Geonames is special, considering the property restriction used in the ontology for codes.

Table 3 summarizes the result of the alignment process between GeOnto and the existing vocabularies/taxonomies. All the resources of this work are available at `http://semantics.eurecom.fr/datalift/tc2012/`.

Vocabulary	**#Classes**	**#Aligned Classes**
LGD	`owl:Class`:1294	178
DBpedia	`owl:Class`:366	42
Schema.org	`owl:Class`:296	52
Geonames	`skos:ConceptScheme`:12 `skos:Concept`:699	– 287
Foursquare	359	46
Google Place	126	41
bdtopo	`owl:Class`:237	153

Table 3. Results of the alignment process between GeOnto and existing vocabularies/taxonomies.

In general, we obtain good results with Silk, with precision beyond 80%: Google Place: 94%, LGD: 98%, DBpedia: 89%, Foursquare: 92% , Geonames: 87% and bdtopo: 92%. We obtained a precision of only 50% with schema.org due to numerous fine-grained categories that are badly aligned (e.g. `ign:Berge owl:equivalentClass schema:Park`).

6 Challenges

6.1 GeoSPARQL

OGC has adopted the GeoSPARQL standard to support both representing and querying geospatial data on the Semantic Web. The standard document [7] 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. 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; (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 (iv) 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. Each of the components described above has associated requirements. Concerning the vocabulary requirements, Table 4 summarizes the seventeen requirements presented in the GeoSPARQL draft document.

Geographic 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 4. Requirements and implementations for vocabulary definitions in GeoSPARQL.

Based on the GeoSPARQL requirements, we were interested in comparing some geospatial vocabularies[19] 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 5.

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 6 shows which triple stores can support part of the GeoSPARQL standard regarding serialization and spatial functions.

6.2 Some Recommendations

The alignment of GeOnto provided in the previous section 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

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

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 5. Comparison of some geo-vocabularies with respect to the GeoSPARQL requirements.

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 taken depending on the technological infrastructure (e.g: data storage capacity, further reasoning on specific points on a complex geometry).

- **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 4). 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[20] 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`.

[20] `http://www.vocamp.org`

– **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 6, `Open Sahara`[21], `Parliament` [22], `Virtuoso`[23] are WKT/GML-compliant with respectively 23 and 13 functions dealing with geodata.
– **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 in the extended version of GeOnto. The choice of the triple store (e.g.,Virtuoso vs Open Sahara) is not really an issue, as the IndexingSail[24] service could also be wrapped on-top of Virtuoso to support full OpenGIS Simple Features functions[25].

Triple store	WKT-compliance	GML-compliance	Geometry supported	Geospatial Functions	GeoVocab
Virtuoso	Yes	Yes	Point	13 functions	W3C Geo + Typed Literal
Allegro-Graph	-	–	Point	3 functions	"strip" mapping data
OWLIM-SE	–	–	Point	4 functions	W3C Geo
Open Sahara	Yes	Yes	Point, Line, Polygons	23 functions	Typed Literal
Parliament	Yes	Yes	Point, Line, Polygons	23 functions	GeoSPARQL vocabulary

Table 6. Triple stores survey with respect to geometry types supported and geospatial functions implemented.

7 Conclusions and Future Work

We have presented in this paper a first step towards interoperability of French geodata in the Semantic Web. The survey of existing modeling of points of interest and geometry shows the different vocabularies and modeling choices used

[21] `http://www.opensahara.com`
[22] `http://geosparql.bbn.com`
[23] `http://www.openlinksw.com`
[24] `https://dev.opensahara.com/projects/useekm/wiki/IndexingSail`
[25] `http://www.opengeospatial.org/standards/sfs`

to represent them. In France, there is a currently a joint effort to publish geographic information in RDF and interlink them with relevant datasets. GeOnto is an ontology describing geospatial features for the French territory. We have proposed to align GeOnto with other popular vocabularies in the geospatial domain. We have used Silk for schema mapping and we have evaluated the results. We studied how to extend the model to take into account efficient modeling for complex geometries. By doing so, we revisited current implementations of geovocabularies and triple stores to check out their compatibility with respect to the new GeoSPARQL standard . We finally made some recommendations and advocate for the reuse of the NeoGeo ontology within GeOnto to better address the IGN requirements. Our future work includes the conversion and publication of a large RDF dataset of geographic information of the French territory together with alignments with other datasets at the instance level.

Acknowledgments

This work has been partially supported by the French National Research Agency (ANR) within the Datalift Project, under grant number ANR-10-CORD-009.

References

1. S. Auer, J. Lehmann, and S. Hellmann. LinkedGeoData - Adding a Spatial Dimension to the Web of Data. In *International Semantic Web Conference (ISWC'09)*, 2009.
2. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5:1–22, 2009.
3. A. de León, L. M. Vilches, B. Villazón-Terrazas, F. Priyatna, and O. Corcho. Geographical linked data: a Spanish use case. In *International Conference on Semantic Systems (I-SEMANTICS'10)*, Graz, Austria, 2010.
4. J. Goodwin, C. Dolbear, and G. Hart. Geographical Linked Data: The Administrative Geography of Great Britain on the Semantic Web. *Transactions in GIS*, 12:19–30, 2008.
5. K. Janowicz, S. Schade, A. Bröring, C. Kessler, C. Stasch, P. Maué, and T. Diekhof. A transparent semantic enablement layer for the geospatial web. In *Terra Cognita Workshop*, 2009.
6. S. Mustière, N. Abadie, N. Aussenac-Gilles, M.-N. Bessagnet, M. Kamel, E. Kergosien, C. Reynaud, and B. Safar. GéOnto : Enrichissement d'une taxonomie de concepts topographiques. In *Spatial Analysis and GEOmatics (Sageo'09)*, Paris, France, 2009.
7. M. Perry and J. Herring. OGC GeoSPARQL- A Geographic Query Language for RDF Data. In *OGC Implementation Standard, ref: OGC 11-052r4*, 06 2012.
8. J. Salas and A. Harth. Finding spatial equivalences accross multiple RDF datasets. In *Terra Cognita Workshop*, pages 114–126, Bonn, Germany, 2011.
9. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and Maintaining Links on the Web of Data. In *International Semantic Web Conference (ISWC'09)*, 2009.

Transforming Between UML Conceptual Models And OWL 2 Ontologies

Jesper Zedlitz[1] and Norbert Luttenberger[2]

[1] German National Library for Economics
[2] CAU Kiel

Abstract. The ISO 19103 standard—defining rules and guidelines for conceptual modeling in the geographic domain—has deliberately chosen the Unified Modeling Language (UML) as "conceptual schema language" for geographic information systems. From today's perspective—i.e. when taking into account today's mature semantic web technology—another language might also be envisioned as language for specifying application-oriented conceptual models, namely the Web Ontology Language OWL 2. Both language definitions refer to comparable meta-models laid down in terms of OMG's Meta Object Facility, but in contrast to UML, OWL 2 is fully built upon formal logic which allows logical reasoning on OWL 2 ontologies. In this paper, we investigate language similarities and differences by specifying and implementing the transformation on the meta-model level using the QVT transformation language.

Keywords: OWL 2, UML, conceptual modeling, ontology, model transformation, GML, Semantic Web, QVT, meta-modeling

1 Introduction

In its introduction, ISO Standard 19103 states: "Standardization of geographic information requires the use of a formal CSL [(conceptual schema language)] to specify unambiguous schemes that can serve as a basis for data interchange and the definition of interoperable services." In focusing on "the combination of the Unified Modeling Language (UML) static structure diagram with its associated Object Constraint Language (OCL)"—a combination, which is probably the most often used CSL—ISO standard 19103:2005 follows mainstream. To illustrate its use, Fig. 1 shows a UML class diagram taken from the Geography Markup Language (GML) standard, where it serves as conceptual model for some application-specific purpose. Advantages are obvious: UML's graphical syntax lets also non-computer scientists easily comprehend the intention of such diagrams. Also in favor of UML is the rich tool support for UML class diagrams which recommends UML as a good starting point for software development.

Unfortunately, UML class models are not completely backed up by formal logic, and we do not enjoy reasoning support as we do for ontologies. The OWL 2

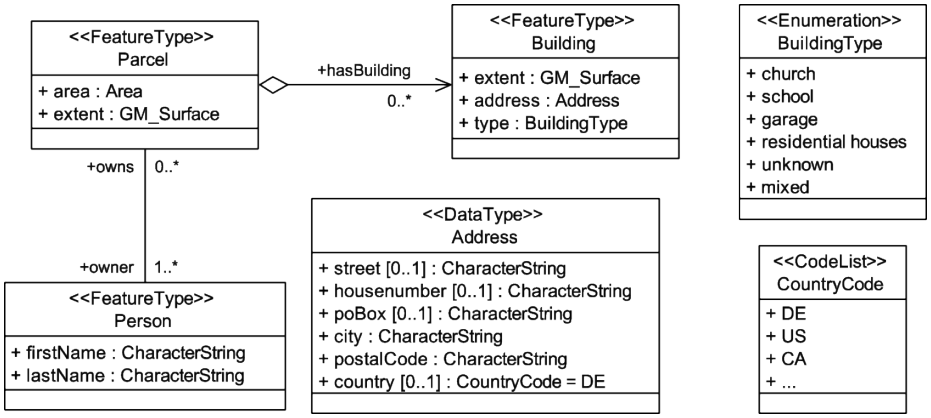


Fig. 1. Example for a UML conceptual model taken from the GML specification [6].

Web Ontology Language[3] or suitable subsets thereof in contrast are completely backed up with formal logical and there is out-of-the-box reasoning support of OWL ontologies. Reasoning over ontologies can be used to discover inferences not detected by programmers, among them subsumption relations between classes and properties in the ontology schema, which helps to determine where a concept can be located in a class hierarchy. Reasoning also helps to assert the consistency of the conceptual model (e.g. validity of intentional definitions or in other words: class satisfiability), and it allows us to inspect the conceptual knowledge encoded in the model. (Bucella et al. [1] follow the same line of arguments when giving their outline for an integration tool for geographical schemas.)

Having given this background, we feel that the following "What-if" question suggests itself: What if OWL 2 were taken as CSL for geographic information systems? Three further arguments back up the validity of our question:

A closer look at conceptual models reveals the systematic use of the object-property model that "has been the basis of the GML encoding model since the first version was adopted by OGC" [6]. It is deeply elaborated in [9]. Needless to be mentioned here, the object-property modeling pattern is at heart of RDF[4], RDF Schema[5], OWL[6], and OWL 2.[7]

Secondly, OWL is one of the building blocks of the semantic web and the Linked Open Data (LoD) Cloud, even if Jain, Hitzler et al. argue that currently the LoD cloud is missing conceptual descriptions [10]. The integration of geographical information systems with the semantic web is an obvious necessity—and might profit much from being built upon common concepts and languages.

[3] http://www.w3.org/TR/owl2-syntax/
[4] http://www.w3.org/TR/rdf-concepts/
[5] http://www.w3.org/TR/rdf-schema/
[6] http:/www.w3.org/TR/owl-ref/
[7] GML even contains a references to RDF: "[...], GML follows RDF (W3C, 1999) terminology [...]"[6, p. 20]

Thirdly, both UML and OWL 2 refer to comparable meta-models laid down in terms of OMG's Meta Object Facility. Thus replacing UML by OWL 2 seems to be a feasible task.

We have chosen a special approach to examine the question if OWL 2 can be used for conceptual modeling. Instead of looking at a bunch of examples (which is always problematic because you cannot be sure to cover all relevant cases with your examples) we approach the question by trying to transform between UML class models and OWL ontologies automatically (in both directions). This systematic approach will show what is possible and what is not.

This paper shows the transformation between a UML model and a OWL 2 ontology with special care for restrictions and extensions GML applies to UML models. We specify a transformation using OMG's Query/View/Transformation (QVT) transformation language and the meta-models of UML and OWL 2.

This paper is organized as follow: Section 2 presents the relation of UML and GML and the restrictions resp. extensions it applies. In Section 3 we show some existing work on the transformation of UML and OWL. Section 4 explains our approach in general. Section 5 shows general differences between UML and OWL 2. In section 6 we present some of our transformations en detail. Section 7 gives a short summary of the paper.

2 UML and GML

The ISO 19109 standard [9] defines rules how to create "UML Application Schema" in a common way. Basis for these application schemas is the General Feature Model (GFM). However, the GFM only defines the semantics of the meta-model but does not provide a concrete syntax how to write the schemas. In ISO 19103 [8] UML is chosen as "conceptual schema language". By defining rules for the usage of UML a so called "UML profile" is defined.

The restrictions made by ISO 19103 limit the number of UML model elements and their use. Also defined are extensions—particularly noteworthy are the stereotypes ≪CodeList≫ and ≪Union≫. However, these are not without controversy, as can be seen below. The ISO 19136 standard—GML [6] picks up the restrictions and amplifies them to some extent. A complete list of restrictions can be found in the GML specification [6].

- All UML elements have the visibility "public". [6, E.2.1.1.1]
- Class names within a class diagram are unique. [6, E.2.1.1.2].
- Operations are ignored. [6, E.2.1.1.2]
- A class can either be a FeatureType if it is marked with the stereotype ≪FeatureType≫, a DataType if it is marked with the stereotype ≪DataType≫ or an ObjectType—classes without any stereotype. [6, E.2.1.1.2]
- A generalization between two classes is only allowed if both classes are FeatureTypes, both classes are ObjectTypes or if both classes are DataTypes. [6, E.2.1.1.2]
- A generalization between classes must not be marked with a stereotype. [6, E.2.1.1.2]

– Multiple inheritance is not allowed. [6, E.2.1.1.2]
– Every association must have exactly two ends which links to a FeatureType, ObjectType or DataType. [6, E.2.1.1.3]
– Associations must not be marked with stereotypes and must not contain attributes. [6, E.2.1.1.3]

As [4, 2.4.3] observed, the UML profile described in the ISO 19103 standard is not a profile within the meaning of the definition of UML profiles given by the OMG. One reason is that the profile defines two stereotypes for data types that are applied to classes. The two stereotypes ≪CodeList≫ and ≪Union≫ are no semantics conserving specializations. For the transformation of classes marked with the disputed stereotypes this observation, however, plays no role. We will use the newly defined semantic of the marked classes.

3 Related Work

Several publications deal with general transformation of UML models into ontologies. Most of them work on XML serializations using XSLT. [2], [5], [3], and [11] fall into this category.

Milanović [12] describes the transformation of a UML model into a OWL ontology using the Atlas Transformation Language, Höglund [7] uses MOFScript for a transformation to OWL 2. However, the goal of his work is validation of models—therefore additional elements needed for the validation are inserted into the ontology that hinder further use in an information system.

Tschirner et al. [13] describe conversion rules from UML-data models to OWL. They specify four main rules to map UML classes and attributes to OWL-classes and properties. However, the constraints specific model elements (e.g. a Union) impose on the model are not mapped.

4 Basic idea

Commonly model driven architecture uses a four-layer architecture: meta-meta-model (M3), meta-model (M2), model (M1) and instance (M0) layer. OMG's MOF is a standard M3-system with a well-developed suite of software tools.

Instead of transforming elements of a M1-model directly we describe the transformation using elements of the M2-meta-models. By describing the transformation on a higher meta-level the transformation does not depend on the models that are going to be transformed. It only depends on the involved meta-models. This enables an elegant description of the transformation—for example compared to a XSLT-based transformation that works with the concrete syntax of M1-models.

It is very common that additional to one or more concrete syntaxes for a language an abstract syntax exists. For example for OWL 2 has various concrete syntaxes: Functional-Style Syntax, Turtle Syntax, OWL/XML Syntax, Manchester Syntax, etc. By working with the abstract syntax our transformation becomes independent of any particular representation.

We choose OMG's QVT Relations Language for our transformations because it is declarative and works with MOF-based meta-models. The support by the OMG consortium and several independent implementations makes it future-proof.

5 Differences of UML and OWL 2

To assess the usage of OWL 2 as CSL we first take a look at some fundamental differences of UML and OWL 2 and point out ways to circumvent some of them.

5.1 Open-World vs. Closed-World Assumption

In UML class models we work under a Closed-World Assumption (CWA): All statements that have not been mentioned explicitly are false. In contrast OWL 2 uses an Open-World Assumption (OWA) where missing information is treated as undecided. These different semantics make it necessary to add various restrictions to the ontology during the transformation process from a UML model to an OWL 2 ontology to preserve the original semantics of the model.

5.2 Profiles

UML has the concept of "profiles" which allow extensions of meta-model elements. There is no corresponding construct in OWL 2. In most cases UML profiles are used to define stereotypes to extend classes. The information of these stereotypes can be mapped to OWL 2 by clever creation of some new classes and generalization assertions. However a large part of an UML profile is too specific and would require transformation rules adapted for the particular profile.

5.3 Abstract Classes

Abstract classes can not be transformed into OWL 2. If a class is defined as abstract in UML no instances of this class (objects) can be created. In contrast OWL 2 has no language feature to specify that a class must not directly contain any individual. An approach to preserve most of the semantics of an abstract class is the usage of a `DisjointUnion`. This would ensure that any individual belonging to a subclass would also belong to the abstract superclass. However, it does not prohibit to create direct members of the abstract superclass.

5.4 Access Control and Operations

In UML the visibility of model elements can be reduced by marking them as "public", "private", etc. It is also possible to declare UML model elements as read only. OWL 2 does not have this kind of control mechanism to restrict the access to model elements. OWL 2 ontologies also do not contain any operations. However, in the list of restriction show in section 2 we have seen that both access control and operations are ignored.

5.5 Global Properties

In OWL 2 it is possible to define (object) properties at ontology level. Connections to classes (in the form of domain and range definitions) are optional. The following listing shows both cases:

```
1 Declaration( ObjectProperty( :belongsTo ) )
2 Declaration( ObjectProperty( :owns ) )
3 ObjectPropertyDomain( :owns :Person )
4 ObjectPropertyRange( :owns :Parcel )
```

`owns` is a connection between individuals belonging to the classes `Person` and `Parcel`. No domain or range has been specified for `belongsTo`, therefore the default value of `owl:Thing` is used for domain and range. The `belongsTo` object property can be used to connect any two individuals because every individual belongs to the class `owl:Thing`.

UML offers two ways to connect classes: class-dependent attributes and associations. As the name states, class-depended attribute belong to a class and connect it with an other class or data type. Associations are package level elements themselves. However, they need (at least) two so called members-ends which require classes as types. Therefore UML associations are not completely suitable to represent (generic) object properties.

5.6 Complement

In many places OWL 2 allows you to work with the complement of classes and data types. In UML, this is not generally possible.

6 Transformations

Several transformation rules for the transformation direction UML → OWL 2 can be found in our article [14] where we have first presented our idea to use a declarative transformation language on meta-model level for transforming generic UML class models into OWL 2 ontologies. However, to answer the question whether OWL 2 could be used as CSL for geographic information systems special challenges of ISO19100/OGC conceptual models have to be taken into account (e.g. available top-level-classes and stereotypes with extended semantics). Therefore we would like to highlight these areas:

6.1 Global Properties

One way to map object properties with `owl:Thing` as domain and range to UML is the definition of a single top-level class that is super-class C_{super} of all other classes in the model (like the `AbstractGMLType`) that represents the `owl:Thing` class. In that case a generic object property can be mapped onto a UML association with two members ends of type C_{super}.

6.2 Complement

As already mentioned the definition of a complement which is possible in OWL 2 is difficult. Only if you define a single top-level class, a *GeneralizationSet* marked *disjoint* can be used to model a class C and its complement ¬C.

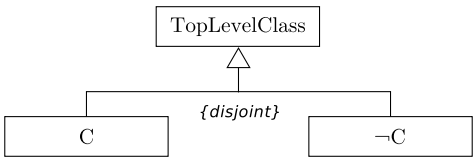


Fig. 2. UML diagram showing how the complement of a class C can be modelled.

In GML such a single top-level class exists: `AbstractGMLType`. Since each object can only be instances of either `FeatureType`, `DataType` or `ObjectType` it would also be possible to use these three classes as *TopLevelClass* when dealing with complements.

For conceptual models following GML's restrictions it is not even necessary to mark the generalization set as *disjoint*. GML does not allow an object to be instance of more than one element type.[6, E.2.1.1.2]

6.3 Associations and Class-Dependent Attributes

In UML two ways to connect classes exist: associations and class-dependent attributes. In the UML meta-model both kinds are represented by the model element *Property* as shown in Fig. 6.3. In general an association can connect two or more objects (cardinality 2..*). However, GML restricts associations to have exactly two ends. That means both class-dependent attributes and associations are connections between two element types. Therefore it stands to reason that the transformation of both associations and attributes can be handled together.

Since the model element *Association* is a subclass of *Classifier* all associations in a UML class diagram are direct members of a package. Therefore an OWL 2 concept that is similar to a association is an object property which is also direct members of an ontology. Associations can be directed or bi-directional. A directed association can be transformed into one object property. For a bi-directional association two object properties will be created—one for each direction. To preserve the information that both resulting object properties were part of one association a `InverseObjectProperties` axiom is added to the ontology.

The transformation of class-dependent attributes is more complex. There are no directly corresponding concepts in OWL 2 that allow a simple transformation. The main problem is that classes in OWL 2 do not contain other model elements which would be necessary for a direct transformation. The most similar concepts in OWL 2 for class-dependent attributes are object properties and data properties.

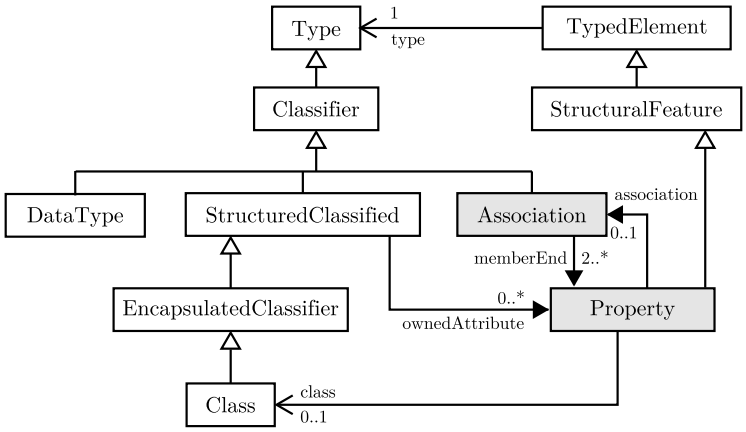


Fig. 3. Excerpt from the UML meta-model showing both possibilities to connect classes.

In both cases the decision whether a *Property* is transformed into an object property or a data property depends on the *type*-association of the *Property*: If it is associated with an instance of *Class* an object property is needed. If it is associated with an instance of *DataType* a data property is needed.

The OWA would allow that two properties that have been transformed from distinct UML properties are interpreted as one. To avoid that and to map UML's CWA best we mark all properties that are not in a generalization relationship (i.e. a `SubPropertyOf` axiom exists for them) as disjoint. To do this we add `DisjointObjectProperties` and `DisjointDataProperties` axioms to the ontology: For all pairs of UML *Property* elements we check if they were transformed into a OWL 2 property, are not identical, no generalization relationship exists between them, and they have not been marked disjoint before.

6.4 Codelist

A *Codelist* is a special kind of enumeration defined by the ISO 19103 standard. A class that is a *Codelist* is marked with the stereotype ≪Codelist≫. In addition to the fixed values of a normal *Enumeration* a *Codelist* might contain other values, too. The GML standard specifies the lexical form of these additional entries.

Similar to the mapping of an *Enumeration* a *Codelist* can be transformed to OWL 2 by using `DataUnionOf` to add the additional values of the *Codelist* to the `DataOneOf` element that has been created for a normal *Enumeration*. The `DataTypeRestriction` element allows an elegant way to add the restrictions for additional values to the data-type—we can use the same syntax from XML Schema[8] that is used in the GML standard.

[8] http://www.w3.org/TR/xmlschema-2/

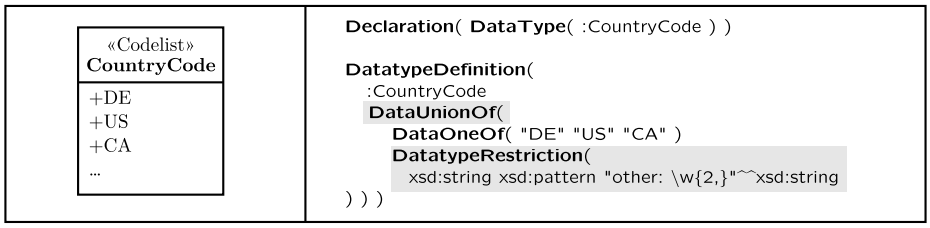


Fig. 4. Transformation of a GML Codelist.

6.5 Union

Another GML specific stereotype is *Union*. The semantics of a *Union* is that only one element of a set of properties may be present at any time. In UML a *Union* is modelled as a class annotated with the ≪Union≫ stereotype. The set of properties is the collection of class-dependent attributes.

We have developed two different mappings to transform a *Union* to OWL 2. The first solution works if the type of all attributes are either data-types or classes. In that case the transformation of the attributes results in `ObjectProperty` or `DataProperty` elements, not a mixture of both.

Let C be a class representing a *Union* with properties $p_1 \dots p_n$. To assure that only one property $p_x \in p_1 \dots p_n$ is specified for an individual we insert a helper property p_{Union} with the domain C and $p_i \sqsubseteq p_{\text{Union}} \forall i \in 1..n$

Now we can add a `DataExactCardinality` axiom to the ontology which limits the use of our helper property p_{Union} to exactly one per individual of class C. That prohibits the use of two or more different properties. However, due to the OWA we cannot make sure that at least one property is present—there might be an individual that is simply not listed in the ontology.

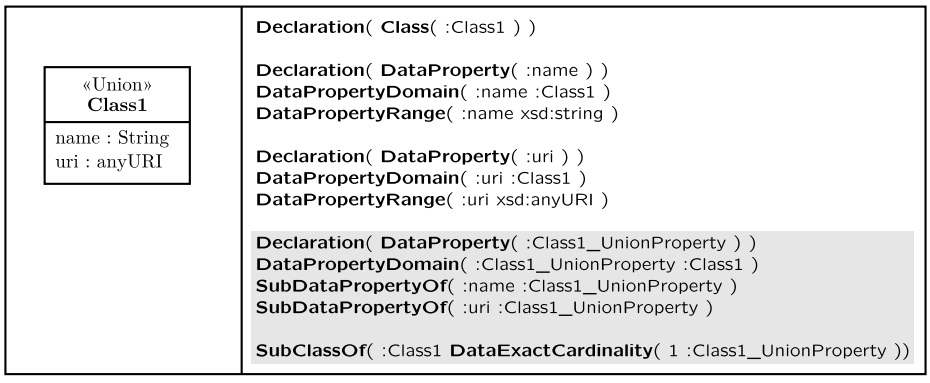


Fig. 5. First solution for a transformation of a GML Union.

The second solution also works with a mixture of `ObjectProperty` and `DataProperty` elements. However, the resulting ontology becomes a bit more complex.

For each property $p_i \in p_1 \dots p_n$ of the union we create a helper class C_i. With a disjoint classes axiom we state that all of these classes are pairwise disjoint: `DisjointClasses(`C_1 ... C_n`)`. Each class is stated as equivalent to a set that contains all those individuals connected by p_i with exactly one individual/literal: `EquivalentClasses(` C_i `DataExactCardinality( 1` p_i `) )` resp. `EquivalentClasses(` C_i `ObjectExactCardinality( 1` p_i `) )`

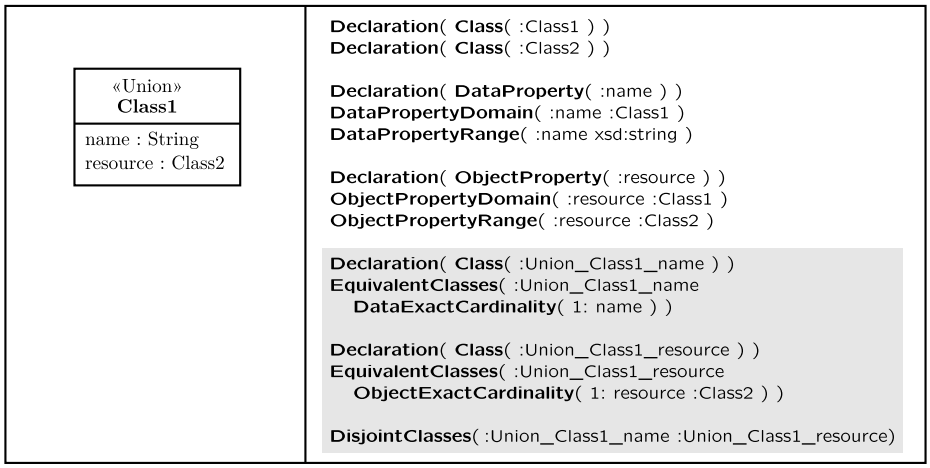


Fig. 6. Second solution for a transformation of a GML Union.

While the first solution only adds $(n + 3)$ axioms per UML property of the union to the ontology the second solution requires $(2n + 1)$ additional axioms per property. Therefore it is clever to choose the first solution if all attributes of a union only link to data-types resp. classes and only choose the second solution if it is a mixture of both.

6.6 Stereotypes

In UML stereotypes can be applied to classes (and other model elements). A few stereotypes are defined in the UML specification. A user can define his own stereotypes in UML profiles. One of the advantages of QVT is the possibility to access profiles and stereotypes.

As mentioned earlier, some of ISO 19103's stereotypes modify the semantics of the model element they are applied to. In that case a modified transformation is necessary. We have shown these specialized transformation above for the stereotypes ≪CodeList≫ and ≪Union≫.

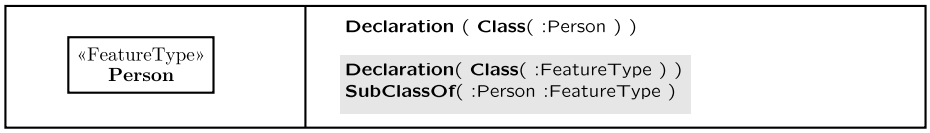


Fig. 7. Transformation of the stereotype ≪FeatureType≫.

For the other three stereotypes defined by ISO 19103 (≪FeatureType≫, ≪ObjectType≫, and ≪DataType≫) OWL 2 classes are defined in the ontology. Classes to which these stereotypes have been applied to become sub-classes of those classes in the ontology. Fig. 7 shows an example of such a transformation.

The transformation of stereotypes into classes in OWL 2 is reasonable since we can connect additional axioms with these classes. That allows us to write down some of the semantics of GML/ISO 19103 in a machine interpretable way:

[6, E.2.1.1.2] states that each element type must be either a *FeatureType*, a *DataType* or an *ObjectType*. There are exclusively these three groups of element types. This can be expressed with a `DisjoinUnion` axiom:

```
1 Declaration( Class( gml:DataType ) )
2 Declaration( Class( gml:FeatureType ) )
3 Declaration( Class( gml:ObjectType ) )
4 DisjointClasses( owl:Thing gml:DataType gml:FeatureType gml:
    ObjectType )
```

Both *FeatureType* and *ObjectType* have a unique identifier[9]. In constract, *DataType* must not have such an identifier. This can be expressed in OWL by defining a `DataProperty` like this:

```
1 Declaration( DataProperty( gml:id ) )
2 DataPropertyDomain( gml:id ObjectUnionOf( gml:FeatureType gml
    :ObjectType ) )
3 DataPropertyRange( gml:id xsd:string )
4 FunctionalDataProperty( gml:id )
```

Since all classes to which the stereotype ≪FeatureType≫ or ≪ObjectType≫ had been applied to in the UML class diagram become sub-classes of either `FeatureType` `ObjectType` in the ontology the data property `gml:id` can be used for them. Instances of classes which had the stereotype ≪DataType≫ applied must not use the key: The domain of the property is `FeatureType` or `ObjectType` and these classes are disjoint with the class `DataType`.

[9] "Object types are types where the instances shall have an identity, [...]" [6, E.2.1.1.2]

7 Summary

We have shown differences and similarities between UML conceptual models following the ISO19100/OGC guidelines and OWL 2 ontologies. The use of QVT Relations Language enables us to describe the transformations between both technology spaces declaratively and to use model elements of the meta-models.

In further work the ideas presented here could be used for a real-world geographic information system that makes use of semantic web technology. Using our transformations the implementation could be based on either an existing UML conceptual model or a newly created OWL 2 ontology or even using alternate editing in UML and OWL 2.

References

1. Buccella, A., Gendarmi, D., Lanubile, F., Cechich, A., Colagrossi, A.: Ontology-Driven Generation of a Federated Schema for GIS. Semantic Web Applications and Perspectives p. 31 (2007)
2. Cranefield, S.: Networked knowledge representation and exchange using UML and RDF. Journal of Digital information 1(8) (2001)
3. Djurić, D.: MDA-based ontology infrastructure. Computer Science and Information Systems 1(1), 91–116 (2004)
4. Eisenhut, C., Kutzner, T.: Vergleichende Untersuchungen zur Modellierung und Modelltransformation in der Region Bodensee im Kontext von INSPIRE (Sep 2010)
5. Gašević, D., Djuric, D., Devedzic, V., Damjanovi, V.: Converting UML to OWL ontologies. In: Proceeding WWW Alt. '04. pp. 488–489. ACM (2004)
6. GML 3.2.1: Geography Markup Language (GML) Encoding Standard 3.2. 1 (2007)
7. Höglund, S., Khan, A., Liu, Y., Porres, I.: Representing and Validating Metamodels using the Web Ontology Language OWL 2. Tech. rep., D. of Information Technologies, Åbo Akademi University (2010)
8. ISO 19103: Norm ISO/TS 19103:2005 Geographic information – Conceptual schema language. ISO, Geneva, Switzerland (2005)
9. ISO 19109: Norm ISO 19109 Geographic information – Rules for application schema. ISO, Geneva, Switzerland (2005)
10. Jain, P., Hitzler, P., Yeh, P., Verma, K., Sheth, A.: Linked data is merely more data. Linked Data Meets Artificial Intelligence pp. 82–86 (2010)
11. Leinhos, S.: OWL Ontologieextraktion und -modellierung auf der Basis von UML Klassendiagrammen (2006)
12. Milanović, M., Gašević, D., Guirca, A., Wagner, G., Devedžić, V.: On Interchanging Between OWL/SWRL and UML/OCL. In: Proceedings of 6th Workshop on OCL for (Meta-) Models in Multiple Application Domains (OCLApps). pp. 81–95 (2006)
13. Tschirner, S., Scherp, A., Staab, S.: Semantic access to INSPIRE. Terra Cognita Workshop (2011)
14. Zedlitz, J., Jörke, J., Luttenberger, N.: From UML to OWL 2. In: Proceedings of Knowledge Technology Week 2011. Springer (2012)

OnGIS: Ontology Driven Geospatial Search and Integration

Marek Šmíd and Zdeněk Kouba

Faculty of Electrical Engineering, Czech Technical University in Prague
smidmare@fel.cvut.cz,kouba@fel.cvut.cz

Abstract. Helping non-expert users to query over complex spatial data requires abstracting from GIS services and operations. This paper introduces an abstraction layer based on OWL ontologies. It provides an intuitive searching GUI for exploring data semantically integrated from different sources. The language OWL 2 QL has been chosen as it performs open-world semantic web reasoning suitable for data integration, while having good computational properties. To support spatial constraints, a custom OWL 2 QL reasoner has been designed as part of this work. It allows relating different objects by means of spatial joins. A prototype system backed by the custom reasoner has been tested on the urban planning domain.

Keywords: Geospatial semantics, OWL 2 QL, Data integration

1 Introduction

The problem OnGIS tries to solve is a semantic search over spatial data, which is motivated by the need of presenting GIS services to non-expert users (with neither technical nor GIS background) in a simple way, still allowing him/her to perform complex queries. The use case at the end of this section shows that it can help a user to obtain relevant information from a complex GIS server, which would be otherwise difficult without understanding the GIS domain.

Another advantage of using semantic technologies for the search is that it can be crafted to work with multiple data sources, which allows for filtering by relations between objects from different sources.

Semantics of the queries is supported by an OWL 2 QL [1] reasoner (see Section 3), which supports e.g. sub-class, sub-property, domain and range axioms. For example, when there is an axiom saying that class *Places of Worship* in an ontology has a sub-class *Churches* in another ontology, it helps to find instances of *Churches* from the latter ontology, when a user asks for instances of the *Places of Worship* class.

A more complex example proving that OWL 2 QL goes beyond RDFS[1] expressivity consists of the following two axioms: *Churches* is a sub-class of the range of the *marriageTakesPlaceIn* object property, and *marriageTakesPlaceIn*

[1] http://www.w3.org/TR/rdf-schema/, cit. 24.7.2012

is a sub-property of the inverse of the *hostsSocialEvent* object property. Then, asking for all places that host social events (i.e. querying the domain of *hostsSocialEvent*) retrieves also the instances of *Churches*.

The main advantage of OWL 2 QL over OWL 2 DL is its tractability, which allows for storing instances in a relational database and posing ontological queries[2] by means of query reformulation to SQL[3]. This makes it possible to store large amounts of data, which is typical in GIS systems, and reason over them. Still, OWL 2 QL is quite powerful (has class and role hierarchies, limited negation, etc.) and it keeps the open world assumption. It is also well settled as a W3C standard.

A motivation, where OnGIS can be used, is to visualize data and maps of the department of urban planning of Prague (the capital of the Czech Republic), being a part of URM (Útvar rozvoje hlavního města Prahy[4]) — City Development Authority of Prague.

URM collects many data sets obtained from various government institutions, taking care of e.g. pollution, noise, flood risks, and land prices. URM groups these data sets into map layers and services, which are available to the public for various analyses (e.g. setting prices for real estate companies, finding a suitable site to build a house for a family). But for a user, who is not a GIS expert, it is not easy to search in a catalogue of GIS services, to find relevant map layers and to work with them.

Thus, the metadata of URM GIS services were extracted (mostly by querying ArcGIS[5] SOAP web services), and stored into an ontology. This ontology was annotated with the OnGIS annotations (see Section 4), making it possible for OnGIS to search it and display the URM GIS services in a map.

2 Related Work

2.1 Linking Ontologies with Databases and Semantic Integration

There are quite many systems for linking ontologies to databases, here are some examples implementing OWL 2 QL or another $DL{-}Lite$ language [2] (see Section 3) that can link to databases or other efficient storages, e.g. QuOnto[6] [3], ROWLKit [4], Mastro [5], OWLIM[7] (OWLIM-SE has spatial support according to [6], but very limited), Stardog[8]. None of the mentioned systems supports (to our knowledge) complex spatial queries.

[2] Queries consisting of elements with defined meanings given by an ontology, and relations between the elements.

[3] This technique of query answering is called Ontology-Based Data Access (OBDA), which uses an ontology as a mediator to access non-semantic data.

[4] `http://www.urm.cz/`, cit. 24.7.2012

[5] A series of GIS software by ESRI, see `http://www.esri.com/software/arcgis`, cit. 24.7.2012

[6] `http://www.dis.uniroma1.it/~quonto/`, cit. 7.10.2011

[7] `http://www.ontotext.com/owlim`, cit. 24.7.2012

[8] `http://stardog.com/`, cit. 24.7.2012

For mapping ontology entities to a database, we have chosen a set of our own simple OWL annotations. It was impossible to reuse existing mapping approaches, e.g. in D2RQ [7], because they do not fit the OnGIS needs, since they do not have some required features (e.g. filtering table results) and they do not allow for needed optimizations (e.g. query containment).

There has been many works on semantic integration ([8] gives an overview). An example is CARIN [9]. The CARIN family of languages combines Horn rules and description logics. It deals with designing a sound and complete inference procedure for answering queries. An application of CARIN, Information Manifold, as presented in [10], serves a similar purpose as OnGIS — information gathering providing uniform access to multiple structured information sources. It uses materialized database views that guarantee accessing only relevant sources. However, Information Manifold does not deal with spatial data sources. C-OWL [11] is an interesting approach for linking semantic data using contextualized ontologies based on OWL.

General database integration tools are not suitable, though some of them support spatial data integration, since they lack a semantic layer.

2.2 GIS Querying Systems

The authors of [12] propose a system for mapping ontology axioms to SQL queries on a database with the focus on geospatial data. Though using ontologies, it does not rely on OWL or any other reasoning. Also it does not focus on multiple data sources.

The system DO-ROAM [13] is quite similar to our OnGIS. It is a web service that focuses on finding places according to activities that a person could perform there. It uses its own OBDA system, which maps ontology concepts and properties to database queries in quite a simple way. The implemented OBDA system is probably not a general OWL reasoner. Again, it does not support multiple data sources, and is not general enough to support reasoning over different domains.

The ontology-based information system in [14] focuses on spatio-thematic query answering for city maps. Its reasoner implements a very expressive logic that has some features OWL 2 DL has not (and vice versa). The system implements its own custom storage, which directly includes the inference algorithms and the query evaluation engine. Spatial data in an ABox[9] are represented e.g. as RCC relations only, or by using a special spatial ABox. But it does not integrate multiple data sources.

The system in [15] links an RDF[10] ontology to databases and WFS[11]. It uses custom rules and algorithms for query rewriting, but it does not provide the standard OWL semantics. However, it supports query answering from multiple data sources, specifically WFS servers for spatial data and databases (via the D2R interface) for attributes.

[9] A set of all ontology axioms about individuals — assertions.

[10] `http://www.w3.org/RDF/`, cit. 24.7.2012

[11] Web Feature Service, an Open Geospatial Consortium (OGC) standard, see `http://www.opengeospatial.org/standards/wfs/`, cit. 24.7.2012

The spatial decision support system in [16] integrates various data sources (OGC standards WMS, WFS, WCS, WPS) and links them with ontologies. It also uses catalogue services via ontologies and automatic web service discovery. However, it focuses more on geospatial analysis and ontology alignment than spatial search.

The authors in [6] use Parliament triple store, supporting geospatial indexes, for storing spatial data and for making complex spatial queries via GeoSPARQL (see Section 3.2) over them. However they use a precomputed data set and do not directly support data integration.

3 OWL 2 QL

OWL 2 QL [1] is a profile of the Web Ontology Language (OWL). The key feature is its tractability (along with other OWL 2 profiles) traded for expressiveness, which is lower compared e.g. to OWL 2 DL. The tractability brings the advantage that description logic queries can be reformulated into SQL and thus RDBMSs (relational database management systems) can be used as OWL 2 QL storage.

OWL 2 QL is based on $DL{-}Lite^{\mathcal{H}}_{core}$, a member of the $DL{-}Lite$ language family defined (in its extended version) in [2], being in turn a member of the description logics family [17]. $DL{-}Lite^{\mathcal{H}}_{core}$ constructs for defining concepts and roles in description logics syntax are:

$$B ::= A \,|\, \exists R\,, \qquad C ::= B \,|\, \neg B\,, \qquad R ::= P \,|\, P^-\,,$$

where A denotes a concept name, B a basic concept, and C a general concept. Symbol P denotes a role name, and R a complex role.

The semantics is defined by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty interpretation domain and $\cdot^{\mathcal{I}}$ is an interpretation function, that assigns to each individual an element of $\Delta^{\mathcal{I}}$, to each concept name a subset of $\Delta^{\mathcal{I}}$, and to each role name a binary relation over $\Delta^{\mathcal{I}}$.

Semantics of the used constructs are defined in Table 1.

Table 1. Constructs used in $DL{-}Lite$ and their semantics

Syntax	Semantics	Comment
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	concept name
P	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	role name
P^-	$(P^-)^{\mathcal{I}} = \{(b,a) \mid (a,b) \in P^{\mathcal{I}}\}$	inverse of a role
$\exists R$	$(\exists R)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b : (a,b) \in R^{\mathcal{I}}\}$	existential quantification
$\neg B$	$(\neg B)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}}$	negation of a basic concept

A TBox[12] can be defined by inclusion axioms of the form: $B \sqsubseteq C$, and $R_1 \sqsubseteq R_2$, interpreted by $\mathcal{I}$ as $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$, resp. $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$.

[12] A set of all ontology terminological axioms — subsumptions of concepts, domains, etc.

An ABox consists of the following assertion axioms: $A(a)$, and $P(a, b)$, where a, b are individuals interpreted by $\mathcal{I}$ as $a^{\mathcal{I}}, b^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The axioms are interpreted by $\mathcal{I}$ as $a^{\mathcal{I}} \in A^{\mathcal{I}}$, resp. $\left(a^{\mathcal{I}}, b^{\mathcal{I}}\right) \in P^{\mathcal{I}}$.

OWL 2 QL extends $DL{-}Lite$ with various features not affecting its tractability, e.g. data roles.

3.1 Owlgres

Owlgres[13] [18] is an open source Java implementation of a $DL{-}Lite^{\mathcal{H}}_{core}$ reasoner developed by Clark & Parsia[14], backed by a DBMS for persisting data (it is tailored to work with PostgreSQL[15] databases).

The original database schema used in Owlgres has two sets of database tables, one for TBox, with one table listing all classes, one for all object properties, etc., and one set for ABox. This original schema for ABox stores all class assertions in one table, all object property assertions in another table, and similarly for data properties and annotations.

3.2 OwlgresMM

In [19], two other schemas are designed, implemented and compared to the original one in Owlgres. The result is that for most cases, storing class assertions (resp. object and data property assertions) in separate tables per named class (resp. named object and data property), which is one of the new schemas, is the most efficient option. OnGIS uses OwlgresMM, which is our extension of Owglres based on this schema.

Another feature of OwlgresMM is that it supports simple annotations for defining how the classes and properties are mapped into a database. But the key quality is it can work with multiple databases. After input query reformulation, it distributes the query among multiple databases that are linked from the imported TBoxes, and it generates the SQL queries to the ones suitable to answer at least a part of the query (i.e. containing data, which can help spatially restricting the query results).

It partially supports GeoSPARQL [20], an OGC[16] standard for geospatial queries and simple geometry manipulation, extending SPARQL[17], a query language for RDF. GeoSPARQL was an inspiration for spatial filters (e.g. within, within distance, and bounding box) and geometry accessors (e.g. geometry, centroid, and area) implementation, with full compliance pending. There are more features implemented, like support for aggregations, various filters, etc.

OwlgresMM is tailored to work with PostgreSQL with PostGIS extension[18].

[13] `http://pellet.owldl.com/owlgres`, cit. 7.10.2011
[14] `http://clarkparsia.com/`, cit. 7.10.2011
[15] `http://www.postgresql.org/`, cit. 24.7.2012
[16] Open Geospatial Consortium, `http://www.opengeospatial.org/`, cit. 24.7.2012
[17] `http://www.w3.org/TR/rdf-sparql-query/`, cit. 24.7.2012
[18] PostGIS (`http://postgis.refractions.net/`, cit. 24.7.2012) adds support for spatial data to PostgreSQL. It allows efficient storage, indexing, and retrieval of geographical data in database tables.

4 Design

OnGIS tries to follow typical web search scenarios. A user is not used to give a search engine a structured query, but instead enters a few keywords, and examines the results the query engine gives back. Then, the user picks the relevant search results, which are added to the list of items to be displayed. This sequence can be repeated several times, until the user fills the list with all the items he/she wants to be displayed.

The front-end part of OnGIS (see Fig. 1) consists of a modular web application. The key aspect of the web application is that it does not depend on any domain specific data structure nor on any technique of obtaining data. The way, how to obtain data, is provided by plugins. Each plugin has to conform to an API, which allows for user query answering, for providing geometries, and for displaying layers. Plugins can support searching over ontology entities, databases, etc., and displaying layers e.g. from PostGIS databases, WMS[19] and ArcGIS servers.

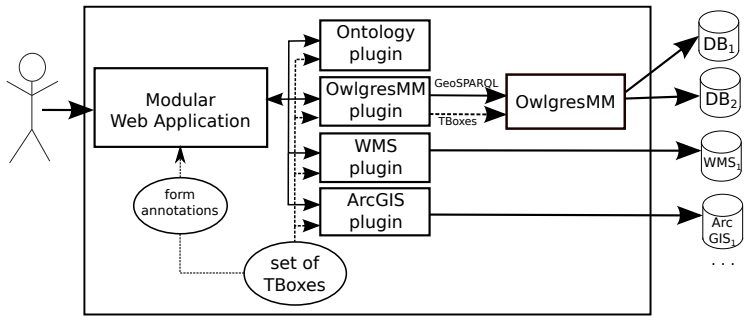


Fig. 1. Overall architecture.

Independence of the web application on domain structure is ensured by using the following specific set of OWL annotations, which serve as an interface between the domain specific ontologies and the web application.

searchable specifies a data property, which should be searched when a user enters a query.

geometry specifies an entity representing spatial geometries — it is useful for spatial queries.

filterable specifies, if a data property is suitable for filtering — it is useful for filtering spatial features (e.g. by attribute queries) to be displayed on a map.

partof specifies relation between entities by an object property, defining part-of relation — it is useful for linking an object to its integral components.

[19] An OGC standard of a simple mechanism for obtaining raster maps, see `http://www.opengeospatial.org/standards/wms/`, cit. 24.7.2012.

priority specifies a numeric priority of entities, which affects the order in which the entities should appear in the search result list.

displayable specifies an entity, which can be displayed on a map as a layer. It may have annotation sub-properties, that are handled by different OnGIS plugins, e.g. for Postgis, WMS, ArcGIS, and ArcGIS RESTful map servers.

An example of the annotations used on the URM domain is illustrated in Fig. 2. Only a part is displayed, what is missing is e.g. the *searchable* annotation marking the URM domain data properties "hasKeywords", "hasDefinition", "hasDescription", etc.

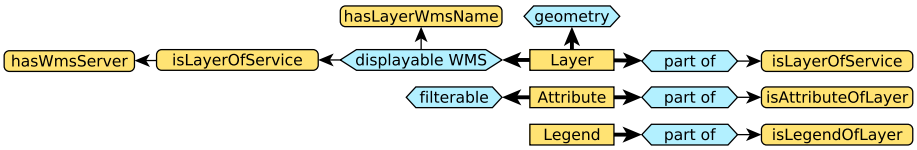


Fig. 2. Illustration of part of OnGIS annotations (the blue diamonds) on URM domain terms (all yellow rectangles). The thick arrows represent annotating, the thin arrows point to annotation values (which are round rectangles). E.g. the "part of" annotation links to an object property, which can be used to relate instances of the annotated class to its parts.

A key advantage of OnGIS compared to other similar systems is that it can spatially relate search results. The relations between objects can be entered in two ways:

simple The simple way is to add a spatial restriction to a search result. Currently, only two restrictions are possible: *inside* restriction and *distance* restriction. The inside restriction filters all other search results, so that they have to be contained inside the search result defining the restriction. The distance restriction is similar — all other search results have to be within the specified distance. These restrictions can be used on more than one search results, but not all plugins need to support multiple restrictions.

by links Another way of specifying relations is by defining links, which is meant for advanced users. A link can be established between two search results, which filters both search results in such a way that they have to be within the specified distance (pair-wise, one feature corresponding to one search result entity, the second feature corresponding to the other search result entity). More link types are to be designed.

A query consists of ontology entities found by the "searchable" annotated properties. In case of the entities annotated as "geometries", in the simple mode, a user can attach the *inside* and the *distance* restrictions to the entities. The semantics of the restrictions is that for each entity attached with one, it restricts

all other "geometry" entities. It is performed in a recursive fashion, e.g. for entities A, B, C, where B, C have some restrictions, then A is restricted by B, which is restricted by C, and also A is restricted by C, which is restricted by B; the cycles in the tree are cut. The restrictions do not apply for "single object" geometries, which are as such reported by OnGIS plugins (not groups of objects, but specific objects, that are directly sought for by the user, and it does not make sense to filter them).

In case of the entities annotated as "filterable", the user can enter filter expressions, which are applied to automatically added (if not already present) entities, of which the "filterable" entities are "part of". The semantics of the links mode is simply restricting only the entities involved in the links. When the query is evaluated, the "geometry" entities annotated with one of the "displayable" annotations are shown on a map.

5 Proof of Concept

An OnGIS prototype has been built as a Java EE[20] application. The prototype provides a web interface for searching and displaying maps using OpenLayers[21].

There are four plugins implemented and used:

Owlapi plugin is used for searching over ontologies via the OWLAPI interface. It performs search only, no layer retrieval nor spatial queries are supported.

OwlgresMM plugin is a connector to OwlgresMM. It receives query requests from the web interface, constructs a SPARQL query and queries appropriate databases using OwlgresMM. It also handles *displayablePostgis* annotation, for which it fetches geometries from appropriate PostGIS databases and generates a layer.

WMS plugin is used for displaying layers in the map from WMS servers. It is driven by *displayableWms* annotation.

ArcGIS plugin is used for displaying layers in the map from ArcGIS servers. It is driven by *displayableArcgis* and *displayableArcgisRest* annotations.

The prototype has been tested with OwlgresMM connecting to two databases, one with OpenStreetMap[22] data, and the other with GeoNames[23] data.

OpenStreetMap is publicly available geographical data of the World. We implemented a special utility, which imports the data to our own spatially enabled database in a suitable format — each category of features in a separate table, and generates an ontology to database tables mapping. As the ontology for OpenStreetMap data, LinkedGeoData[24] ontology was used, which is a part of Linked Data. Its expressivity is quite limited, and it is no problem to fit it into

[20] `http://docs.oracle.com/javaee/`, cit. 24.7.2012

[21] An open-source JavaScript library for displaying and interacting with maps from many raster and vector sources, see `http://openlayers.org/`, cit. 24.7.2012.

[22] `http://www.openstreetmap.org/`, cit. 24.7.2012

[23] `http://www.geonames.org/`, cit. 24.7.2012

[24] `http://linkedgeodata.org/`, cit. 24.7.2012

the OWL 2 QL expressivity. GeoNames is a geographical database of points. However, it has not been used in the example shown below.

The semantics of crowdsourced data, like OpenStreetMap and GeoNames, is not always very precise, compared to the professional data, as from URM, with more accurate semantics. Therefore, it would be useful to employ data available from other local authorities, e.g. cadastral offices.

As an example of many experiments performed (involving various natural objects, ways, city places, etc.), let us search for places of worship with two restrictions: they have to be close to a park (with maximum distance of 100 m), and they have to be inside a specific part of Prague (borough named "Praha 2"). Such queries, integrating the URM services and other data, may help citizens in making their housing decisions. The OnGIS system connects all the data mentioned earlier: URM GIS services, OpenStreetMap and GeoNames data.

The query has to be posed to the system in an understandable way. One way would be to write it as a piece of text with simple rigid structure, with its terms having known meanings. We chose another one, constructing the query iteratively by adding objects in the query (e.g. a place, a group of places, an attribute) one by one. A user enters a keyword for each object in the query first, and from the results found he/she picks the appropriate one, which is added to a list. As an example, we entered the keyword "worship", which found the class "Place of Worship" from OpenStreetMap ontology, that was added to the final query object list, as in Fig. 3. We followed entering the rest of the terms (see the caption of the figure).

Fig. 3 also shows the appropriate filter fields in the list. For example, the *name* attribute is an instance of the class Attributes in the domain specific URM GIS services ontology. This class is annotated with the general OnGIS *filterable* annotation, hence it does make sense to provide the user with text input field for filtering. We entered "Praha 2", the requested borough here. The other objects are either URM GIS layer instances (boroughs) or OpenStreetMap classes (parks and places of worship), which are all annotated as being both displayable (thus they are shown in the map in Fig. 4) and as being geometries (suitable for performing spatial queries). Thus these items contain fields for filtering via maximum distance to them and for filtering other items only to those being inside of them. Therefore, we ticked the inside restriction of boroughs, and entered "100" (meters) to the max. distance restriction of parks.

Fig. 4 contains the results of the query. It is a detail of a map displayed in OpenLayers, having URM technical land usage layers in the background (with the river Moldau on the left-hand side). The colors correspond to the result list in Fig. 3, thus the parks are blue, and the places of worship are red. The display of boroughs was disabled, to make the map more legible.

6 Conclusion and Future Work

The designed ontology based system, OnGIS, is suitable for distributing queries over spatial data to multiple data sources with spatial restrictions among them.

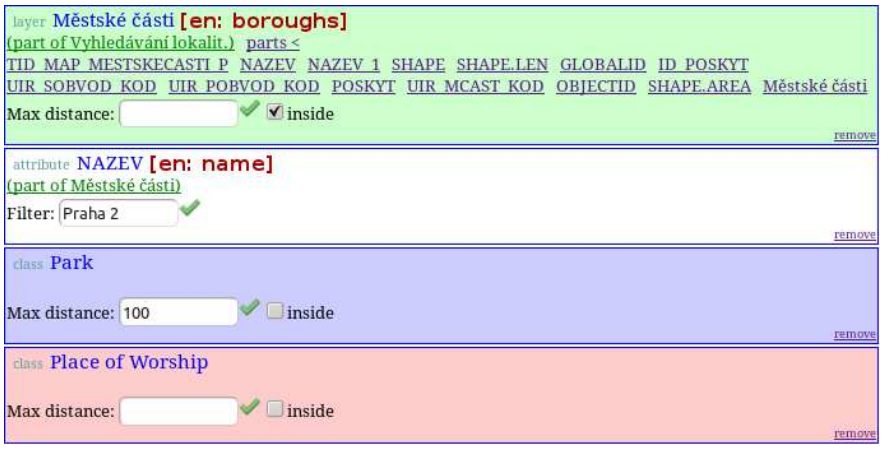


Fig. 3. OnGIS having a few search results added. We are looking for places of worship, thus the "Place of Worship" OpenStreetMap class found by keyword "worship". For the park restriction, we simply sought for "park", which resulted in "Park" OpenStreetMap class. To filter by borough name, it is easier to find boroughs layer first, and then its name attribute (since there are a lot of name attributes of many layers). So we added "Městské části" (meaning boroughs), which is from URM data. Then its parts can be shown in the boroughs item. One of the parts is "název" (meaning name).

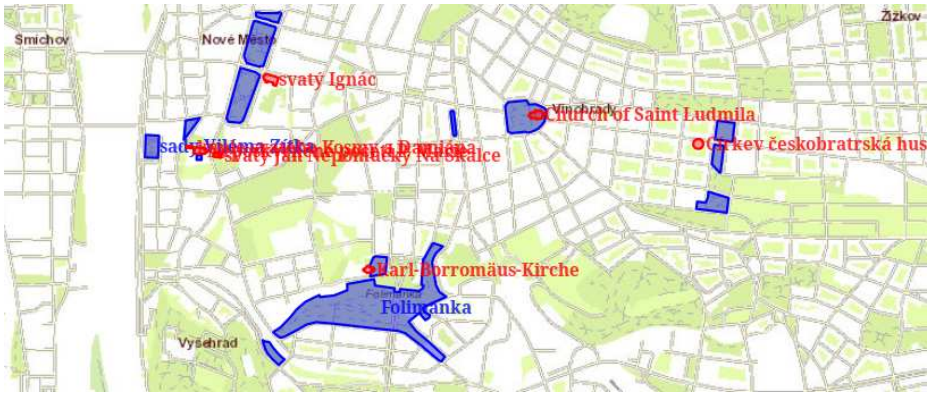


Fig. 4. OnGIS displaying a map with the search results. The blue areas are parks, and the red spots are places of worship (note that they are all close to the parks, as requested by the query), with both appearing only in the borough "Praha 2".

It is based on the idea that the domain-specific data structures linked to the system are described by domain-specific ontologies, that are annotated with general OnGIS annotations, which are the entry points for accessing the data from OnGIS.

A strong aspect of the system is that it can perform filtering by spatial relations between objects — efficiently within one database data source, or even between heterogeneous data sources.

Its prototype supports a few GIS services at the moment: WMS and ArcGIS servers, and spatially-enabled relational databases (specifically Postgres with PostGIS extension).

There are many ways of extending OnGIS. An obvious one is supporting other GIS services via plugins, e.g. plugin for WFS servers (which allow for spatial queries and displaying features). Also a plugin for querying RDF repositories via SPARQL endpoints, possibly supporting GeoSPARQL would help using many data sources (naturally the ones in Linked Data — DBpedia, etc.) would greatly extend the system. The OwlgresMM plugin will be extended to fully support GeoSPARQL to allow compatibility with other systems.

Using non-spatial data would require non-spatial links between objects, which requires an appropriate extension of OnGIS GUI to support it. Also the spatial relation restrictions need extending, with an appropriate semantics designed. Complete query correctness and soundness verification is also necessary.

The current OnGIS query input, iterative addition of single query objects, can be substituted by entering a query via a piece of text with simple rigid structure (an expression not in a natural language, but very restricted) defining objects, and their restrictions and relations. Some users may find it more intuitive and as a quicker way of entering a query. To give the terms in the expression specific meanings, the terms could be picked from a list of suggestions after typing a few first letters (such suggestion lists are widely used by web search engines).

Filtering data based on a spatial relation within a spatially enabled relational database is a computationally expensive operation. Experiments show, that Postgres, when asked to find objects from one table within a distance to objects from another table, uses sequential scan on one table, and spatially indexed access on the other (using the index for object from the sequential scan from the former table). This can take a long time for otherwise unrestricted query over large tables. A solution to alleviate the problem may be to implement seeded tree algorithm according to [21].

Acknowledgments. This work was supported by the CTU research funding and the research programme no. MSM 6840770038 funded by the Czech Ministry of Education.

References

1. World Wide Web Consortium: OWL 2 Web Ontology Language: Profiles, OWL 2 QL, http://www.w3.org/TR/owl2-profiles/#OWL_2_QL. (2009)

2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. of Artificial Intelligence Research **36** (2009) 1–69
3. Acciarri, A., Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Palmieri, M., Rosati, R.: QuOnto: Querying ontologies. In: Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005). (2005)
4. Corona, C., Ruzzi, M., Savo, D.F.: Filling the gap between OWL 2 QL and QuOnto: ROWLKit. In: Description Logics '09. (2009)
5. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The Mastro system for ontology-based data access. Semantic Web Journal **2**(1) (2011) 43–53
6. Battle, R., Kolas, D.: Enabling the geospatial semantic web with Parliament and GeoSPARQL. Semantic Web Journal, to appear
7. Bizer, C., Seaborne, A.: D2RQ - treating non-RDF databases as virtual RDF graphs. In: ISWC2004 (posters). (November 2004)
8. Telang, A., Chakravarthy, S., Huang, Y.: Information integration across heterogeneous sources: Where do we stand and how to proceed? In: COMAD, Computer Society of India / Allied Publishers (2008) 186–197
9. Levy, A.Y., Rousset, M.C.: Combining horn rules and description logics in CARIN. Artif. Intell. **104**(1-2) (1998) 165–209
10. Levy, A.Y., Rajaraman, A., Ordille, J.J.: Query-answering algorithms for information agents. In: AAAI-96. (1996)
11. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing ontologies. In: ISWC. Lecture Notes in Computer Science, Springer Verlag (October 2003) 164–179
12. Baglioni, M., Masserotti, M.V., Renso, C., Spinsanti, L.: Improving geodatabase semantic querying exploiting ontologies. In: GeoSpatial Semantics, Springer (2011)
13. Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T., Rau, R.: DO-ROAM: Activity-oriented search and navigation with OpenStreetMap. In: GeoSpatial Semantics, Springer (2011)
14. Wessel, M., Möller, R.: Flexible software architectures for ontology-based information systems. Journal of Applied Logic – Special Issue on Empirically Successful Computerized Reasoning (2009)
15. Zhao, T., Zhang, C., Wei, M., Peng, Z.R.: Ontology-based geospatial data query and integration. In: GIScience. Volume 5266 of Lecture Notes in Computer Science., Springer (2008) 370–392
16. Zhang, C., Zhao, T., Li, W.: The framework of a geospatial semantic web-based spatial decision support system for digital earth. Int. J. Digital Earth **3**(2) (2010) 111–134
17. Baader, F., Calvanese, D., McGuinness, D.L., Patel-Schneider, P., Nardi, D.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
18. Stocker, M., Smith, M.: Owlgres: A scalable OWL reasoner. In: OWLED. Volume 432 of CEUR Workshop Proceedings., CEUR-WS.org (2008)
19. Šmíd, M.: Using databases for description logics. Master's thesis, Czech Technical University in Prague, Faculty of Electrical Engineering (2009)
20. Open Geospatial Consortium: OGC GeoSPARQL — A Geographic Query Language for RDF Data, http://www.opengeospatial.org/standards/geosparql. (2012)
21. Lo, M.L., Ravishankar, C.: The design and implementation of seeded trees: an efficient method for spatial joins. Knowledge and Data Engineering, IEEE Transactions on **10**(1) (jan/feb 1998) 136 –152

Semantifying OpenStreetMap

Alkyoni Baglatzi, Margarita Kokla, Marinos Kavouras

School of Rural and Surveying Engineering, National Technical University of Athens
H. Polytechniou Str. 9, 15780 Zografos Campus, Greece
alkyoni.baglatzi@gmail.com, (mkokla, mkav)@survey.ntua.gr

Abstract. OpenStreetMap is one of the best examples of Volunteered Geographic Information. Its success relies on the ease of use and the freedom it provides. Users are supposed to geolocate their Points Of Interest and annotate them with a tag. There is no certain vocabulary or ontology of the tags that users have to commit to. The whole tagging process is done in a bottom-up manner in which the community on a wiki basis discusses this issue. Allowing users to use tags freely increases the usability of OpenStreetMap but at the same time causes semantic interoperability problems. What is needed, is a way to structure the tags while satisfying the freedom criterion. As a solution, we suggest the alignment of the tags to well structured top level ontologies. A middle layer approach for bridging the gap between the bottom-up tags of the users and the top level Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is proposed. The idea of "games with a purpose" is utilized to assists non-expert users in aligning their tags to DOLCE.

Keywords: VGI, OpenStreetMap, tagging, semantics, alignment, games with a purpose, top level ontology

1 Introduction

User generated content is an important emerging research area. Engaging the crowd in performing new tasks brings about new opportunities and new challenges. In the geographic domain, the term Volunteered Geographic Information (VGI) was coined by Goodchild [12] for describing the collaborative mapping activities of users and contribution of geographic data. OpenStreetMap which was initiated at University College London (UCL) in July 2004 by Steve Coast, is one of the most pervasive and representative examples of VGI [13].

OpenStreetMap offers an open and easy to use platform that enables contributors to upload geographic information collected from mobile devices or aerial images. The data model is simple and consists of nodes, ways and relations. Each mapped entity is accompanied with a tag. There is no formal ontology or vocabulary of predefined tags that have to be adopted by the users, because as argued by Steve Coast, the founder of OpenStreetMap: "no individual could design such an ontology that would be all-encompassing, and even if they could start no two individuals would agree on it[9]".

Tags that facilitate the annotation of Points of Interest (POIs) in OpenStreetMap come in key-value pairs [1] for instance, `amenity=bar`, `natural=beach`, `landuse= forest`. There is no standardized way on how users shall annotate their POIs neither on the naming level nor on which entities shall be tagged under a certain name.

On a wiki [2] and mailing list [3] basis, the community exchanges opinions about the tags proposing new tags or tags that should be abolished. The most common tags in use, can be looked up in Taginfo [4]. The tags (or Map Features as found in the wiki) are listed in some kind of loose hierarchy.

Although the freedom and openness provided eases the tagging procedure, it causes semantic interoperability problems. User generated content is heterogeneous which leads to ambiguity, redundancy and inconsistency of the tags. As a result, findability of the correct tag for annotating a POI as well as information searching and retrieval is ineffective, an issue that has already been described for instance in [4].

What is needed, is the combination of this loose hierarchy with well structured, organized, formalized top level ontologies and specifically the alignment of users' tags to concepts of a top level ontology. Top level ontologies can be seen as a structured collection of semantic primitives or meta level concepts that are used to further define domain concepts [14]. By aligning the domain concepts to the top level ontologies, the meaning of these concepts gets grounded in the semantic primitives. This universal view of top level ontologies is also a reason why they are more suitable than domain ontologies for the alignment process. The meta level concepts act as reference points in relation to which, the domain concepts are defined.

OpenStreetMap is open to non-expert users of geographic data and thus, the tagging attitude is rather intuitive than based on scientific methodologies and knowledge. This calls for an alignment to a top level ontology, which underlying design principles are prescribed by common sense. As has been argued by its creators, the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [21] "has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human commonsense "([10] p.2). That is the reason why in the present work the DOLCE ontology has been chosen. Specifically, the extension of DOLCE, DOLCE Ultralite [5] (DUL) was regarded as more suitable because it replaces the complicated *endurant, perdurant* division with *object* and *event*. Nevertheless, also other ontologies that satisfy this criterion could have been used instead.

Aligning the tags to the top level ontology in a top-down approach with the aid of knowledge engineers would have been accurate but time consuming. Concerning maintenance, the dynamical nature of OpenStreetMap with the option

[1] We use true type fonts to refer to tags
[2] `http://wiki.openstreetmap.org/wiki/Map_Features`, last accessed 27.07.2012
[3] `http://lists.openstreetmap.org/listinfo`, last accessed 27.07.2012
[4] `http://taginfo.openstreetmap.org/tags`, last accessed 27.07.2012
[5] `http://www.loa-cnr.it/ontologies/DUL.owl`, last accessed 05.09.2012

of new tags being introduced or old removed, would demand the repetition of the alignment procedure very often which is resource inefficient.

Our research question is "How to find a user friendly way to align the tags to top level ontologies?". We aim at defining a methodology that will enable the bottom-up alignment of tags to top level ontologies.

For that, we propose the use of the well established idea of "games with a purpose" [34] applied to OpenStreetMap. Specifically, we choose "question games", a certain type of "games with a purpose", for assisting users in aligning their tags to the concepts of the top level ontology. We use DOLCE Ultralite, an extension of DOLCE to align the tags to.

The contribution of this work is twofold: (a) it provides an analysis of the semantic inconsistencies that emerge from the current state of the tagging process in OpenStreetMap and (b) it proposes a way to combine top-down and bottom-up approaches by preserving the advantages of both that is, the freedom and easiness of the first and the structure, organization of the latter. For that, a "question game" is designed. The main objective is to reuse existing methods in order figure out a methodology for overcoming the semantic inconsistencies of OpenStreetMap.

The remainder of the paper is organized as follows. Section 2 provides information on the related work. Section 3 investigates some semantic inconsistencies that were found in the OpenStreetMap tags. Section 4 describes the proposed methodology for aligning the tags to the top level ontology and Section 5 concludes the paper and discusses possible future directions.

2 Related Work

With the increase of user involvement in the web and the rising amount of user generated content, tagging was introduced for annotating purposes. Flickr [6], del.icio.us [7], citeulike [8] and youtube [9] are just few examples where users added tags to describe their content.

The tagging behaviour has been examined in multifarious ways with several methods in order to understand the commonsense ground of user generated content. Thomas Vander Wal introduced the term folksonomy for describing this collaborative tagging [33]. Clustering methods have been used to investigate in a bottom-up manner kinds of tags people use in their annotations.

Ontology learning has substantially benefited from these kind of studies as they unfolded the way people perceive and use different tags. As a result, ontologies can be designed more efficiently.

The need for combining well structured ontologies with hierarchically loose folksonomies has already been acknowledged. Especially the problem of the missing relations between tags in folksonomies has been addressed in [1]. With the

[6] `http://www.flickr.com/`, last accessed 27.07.2012
[7] `http://delicious.com/`, last accessed 27.07.2012
[8] `http://www.citeulike.org/`, last accessed 27.07.2012
[9] `http://www.youtube.com/`, last accessed 27.07.2012

aid of ontologies, different kinds of relations between tags like subsumption relations, disjointness relations, generic relations, sibling relations and instance of relations were found and added.

Data mining techniques and ontologies are combined in [29] to make the semantics of the tags explicit. Tags are preprocessed, clustered and related to concepts in different ontologies. Swoogle [10] is used as a search engine for finding appropriate ontologies.

In [19] knowledge from folksonomies is extracted with data mining techniques and related to upper ontologies. After a preprocessing step, tags are related to WordNet [11] and enriched with its relations. The methodology is applied to datasets from flickr and citeulike.

Aligning folksonomies to domain ontologies is utilized in [24] for annotating blog posts. The main goal is to limit tags' ambiguity and variation. In the first step, users are free to choose tags for annotating the blog posts. In the second step, ontology concepts are shown to them and in an interactive, semi-automatic manner, users are asked to relate their tags to the concepts from these ontologies. Tags have to be to explicitly matched to concepts from the ontologies.

WordNet is used in [17] to order tags in a hierarchical way. A tool has been built that makes the navigation in tag spaces more comprehensive for users. As a result, browsing and retrieving related tags is made easier and more efficient.

Concerning OpenStreetMap, important research has been conducted in analysing the tagging behaviour of users i.e. which the most edited entities are and how they change over time [22,23]. This provides evidence about the importance of certain entities and the different ways they are perceived by the users broadening the research agenda of user generated geospatial content.

The power of enriching OpenStreetMap with other sources has been demonstrated in the LinkedGeoData project [2]. Instances of OpenStreetMap are published according to the linked data principles and linked to DBpedia[12].

The problem of missing relatedness between geographic entities is addressed in [3]. OpenStreetMap "spatially rich but semantically poor vector dataset" and DBpedia "spatially poor but semantically rich ontology", are combined providing the user with information about a geographic entity. An important contribution is the consideration of map scale in relating concepts to geographic entities.

OSMonto[13] has been developed as an OWL ontology of OpenStreetMap tags [7]. Keys are translated into classes and values into subclasses. The design decision was to be as close as possible to the tagging process enabling querying of the OpenStreetMap database. That is why the tags were adopted as represented in the tag wiki and no conceptual conflicts or ontological mismatches were confronted. OSMonto is used in the DO-ROAM [14] project which aims at expanding

[10] `http://swoogle.umbc.edu/` last accessed 27.07.2012
[11] `http://wordnet.princeton.edu/` last accessed 27.07.2012
[12] `http://wiki.dbpedia.org/About` last accessed 27.07.2012
[13] `https://raw.github.com/doroam/planning-do-roam/master/Ontology/tags.owl`, last accessed 27.07.2012
[14] `http://planning.do-roam.org/`, last accessed 27.07.2012

the search capabilities not only to the POIs but also to the activities that can be performed at a certain location [6].

Scheider et al. [26] argue for the need of more functional or affordance oriented representation of OpenStreetMap tags. They underline the fact that the current tagging practise in OpenStreetMap is not efficient enough for representing the identity of the POIs. As they mention, tagging a cafeteria which is also open at night serving alcohol with `amenity=cafe` may exclude the *alcohol-serving* functionality. They suggest that tags should be grounded in affordances. Accompanying the tags with richer descriptions on the affordances of the POIs, may harden the annotation process but will make the querying process more efficient.

3 Semantic Inconsistencies of OpenStreetMap Tagging

The freedom and easiness of assigning tags to POIs is a hallmark of the success of OpenStreetMap. But, on the semantics of the tags, it leads to several interoperability problems. As this tag collection is a result of a bottom-up user generated effort, it lacks some proper semantic structure. Especially the absence of relations like the hypernymic relation which describe the is-a relation between a concept and its genus and the meronymic relation that is the part-of relation both acting "as the cement that links up concepts into knowledge structures" [15] makes the annotation and searching process cumbersome.

In contrast to geographic ontologies, vocabularies, taxonomies created by domain experts, the key-value pairs are organized in a loose way. Although there is some type of clustering or thematic grouping of the tags in the wiki, which differentiates it from traditional folksonomies where hierarchical information is missing, conceptual inconsistencies still exist. This section aims at providing some examples of the semantic inconsistencies that arise from the tagging strategy of OpenStreetMap.

To start with, there is no common criterion according to which the tags are organized. As a result all keys are in the same hierarchical level. That is, all primary features as listed in the wiki i.e. `amenity`, `aeroway`, `historic`, `landuse`, `manmade`, `craft`, `sport`, `tourism`, `power`, `shop` etc. are treated equally, which results in conceptual vagueness and inconsistency. For instance, the nature of `office` or `shop` is manmade. By not relating these tags to the tag `manmade` with the class-subclass relation, important inheritance information gets lost. Same applies to i.e. `shop`, `office`, `building` which could be subclasses of `amenity`.

On a more sophisticated level, a disadvantage of the flat structure of the tags is the fact that no deeper associations between geographic entities can be established. For instance, in the current tagging procedure there is no way of explicitly stating that within `landuse=commercial`, geographic entities like `shop=bakery`, `office=architect` are located.

Redundancies of tags provide us with evidence about the different conceptualizations of certain POIs that users have. For instance, hotel, hospital, school are tagged as `tourism=hotel`, `building=hotel`, `amenity=hospital`, `building=ho-`

spital, amenity=school, building=school. Users assign the same value (be it hotel, hospital or school), to two different keys namely tourism and amenity.

Another finding is that tags related to activities are used to describe POIs i.e. sport=climbing, sport=basketball, leisure=dance, leisure=fishing. While describing the activity that can be performed at a certain POI, they are used to annotate the POI itself. This may be confusing in terms of information search as it perplexes the geographic entity with its function.

The primary tag amenity also creates confusion since it is used to represent a wide variety of heterogeneous features (e.g., schools, parking lots, bus stations, banks, hospitals, nightclubs, etc.). Although different subcategories of amenities are defined in the wiki (such as sustenance, education, transportation, financial, healthcare, etc.) to further classify different types of amenities, their possible values (such as school, bar, embassy, etc.) directly refer to the general tag amenity, e.g., amenity=school, amenity=bar, amenity=embassy.

The above cases are some examples of the semantic problems caused by a bottom-up, intuitive approach. Furthermore, although the proposed tags have resulted from consensus, they do not necessarily represent a common and wide conceptualization of geographic concepts. For this reason, although such approaches have stimulated considerable interest and resulted in the collection of huge volumes of geospatial data, they are accompanied by problems, such as the creation of arbitrary attributes or attribute values, multiple tags for the same geographic features, disagreement on the name of features, etc. [22]. The present research aims at the design of a "game with a purpose" to align the loose hierarchy of OpenStreet Map tags with a well structured top level ontology to provide meaningful and cognitively important associations between tags.

4 Question Game for Aligning OpenStreetMap Tags to DOLCE Top Level Ontology

4.1 Introduction to the DOLCE Ultralite Top Level Ontology

DOLCE [21] is a foundational or top level ontology developed within the WONDERWEB project[15]. It is an ontology of particulars and comes in different versions DOLCE Lite-Plus [16] and DOLCE+ DnS Ultralite [17] (or DOLCE Ultralite).

In the present work, the DOLCE Ultralite was chosen because its categories and organization is simpler and more intuitive than the other versions as argued by its authors. The top concept *entity* is categorized into *abstract*, *event*, *information entity*, *object* and *quality*. For the alignment, the class *object* and especially its subclasses *physical* and *social object* are of high interest. For further information on the ontology we point the reader to the related links.

[15] http://wonderweb.semanticweb.org/, last accessed 27.07.2012
[16] http://www.loa-cnr.it/ontologies/DLP397.owl, last accessed 05.09.2012
[17] http://www.loa-cnr.it/ontologies/DUL.owl, last accessed 05.09.2012

4.2 Games with a Purpose

"Games with a purpose" were introduced by Luis von Ahn [34] as a way to make use of the human computation for solving complicated tasks. As he argues, machine capabilities are limited in contrast to human reasoning capacities. As a result, there is a need for human involvement. The core idea is that this involvement be easy and motivating for users but at the same time efficient.

Two examples described by Luis von Ahn are the ESP game [35] where users are labelling images in a simple web based game and the Peekaboom [18] game for adding location information to the images. Further examples are the Listen Game for the annotation of music [32] and the Phrase Detectives [19] game for the annotation of text [5].

A detailed collection of different "games with a purpose" can be found in [31]. Also health sciences benefit from "games with a purpose". For instance, the game Foldit was developed for engaging the crowd in protein unfolding [8,11].

In the context of the semantic web, "games with a purpose" are used for building ontologies such as the OntoGame [27]. Similarly to the annotation of images game, wikipedia articles are shown to users who have to evaluate the content of the text and summarize the content on a common way. A wider range of "games with a purpose" for the semantic web can be found in [28].

For ontology alignment, SpotTheLink was designed as a continuation of the OntoGame framework [30]. In this game, users are shown concepts and pictures from DBpedia and then have to agree on one concept from the PROTON [20] ontology.

"Question games" (also seen as "question driven games"), is a type of "games with a purpose" rooted back to the 20q [21] game where the computer tries to guess the concept that a player has in mind based on his/her answers to certain questions. This rational has been successfully used in [16] for ontology engineering purposes and specifically for knowledge acquisition. A knowledge engineer was supposed to ask a domain expert up to 20 questions in order to obtain important concepts and relations that had to be formalized in the ontology. A similar technique has been used for aligning concepts to DOLCE in [18].

4.3 Question Game for OpenStreetMap

The current work, proposes the use of an interactive "question game" for aligning OpenStreetMap tags to the DOLCE Ultralite ontology. The main prerequisite is to hide the complexity of ontology from the users while designing a smart way to align the tags to it. It would have been of little benefit to directly confront users with concepts from the ontology like *information object*, *designed artifact* etc. and ask for the direct alignment of their tags to them.

[18] `www.peekaboom.org`, last accessed 27.07.2012
[19] `http://www.phrasedetectives.org`, last accessed 27.07.2012
[20] `http://proton.semanticweb.org/`, last accessed 27.07.2012
[21] `20q.net/`, last accessed 27.07.2012

The "question game" plays this mediation role between the tags of the non-expert and the well structured and formalized ontology. In such a way, users' freedom of choosing tags is preserved. Anchoring the tags in the top level ontology is catalytic for knowledge sharing and tag reconciliation and disambiguation.

The "question game" is part of the annotation process and its strategy is as follows. Users who want to annotate a certain geographic feature, after deciding which tag they prefer, have to answer some questions in a simple user interface. These questions are simple and rather intuitive in order to be easily understood and quickly answered. Each ontology concept is represented by one question. Examples of these questions can be seen in Table 1.

1. Is it a (physical) object like a river or a stadium?
2. Is it a material, like sand or mud?
3. Is it a boundary of an area like an electoral division?
4. Does it imply some kind of action like swimming or dancing?
5. Can you observe and measure it?
6. Does it have a location?
7. ...

Table 1. Sample questions for the alignment process

The questions refer to the values of the tags. By answering a question, the corresponding key of the tag is aligned as a class to the DOLCE Ultralite concept and the value of the tag as a subclass to the corresponding key. For the first prototype only boolean answers (yes and no) will be required. When a positive answer is given, the tag is aligned to the DOLCE Ultralite concept related to the question. Negative answers, trigger new questions until a positive answer is given. Expected results of the alignment can be seen in Fig. 1.

As can be seen, the keys `tourism`, `building`, `amenity`, `highway` and `leisure` are aligned to the class *Designed Artifact*. That is, they are grouped according to their common criterion. As a result, their scattered listing in the OSM Features wiki is organized facilitating easier findability of the tags and reference of their meaning.

The purpose of the game is to directly align the tags to concepts from the ontology. In this early stage, there is no interaction between users in order to agree on a common tag which is a common practise in games with a purpose. Moreover, the "question game" is not used in order to find out whether tags are used correctly or how consistent they represent the POIs.

For motivating users, well known techniques such as ratings for top users i.e. as seen in [25] or the geo-wiki project [22] are applied. This commonly used technique is documented in [20] as the "glory or recognition motivator".

Concerning the evaluation of the game possible options are usability test measuring the level of ease and fun of the game. Analysis of the aligned tags can

[22] `http://www.geo-wiki.org/login.php?menu=home`, last accessed 27.07.2012

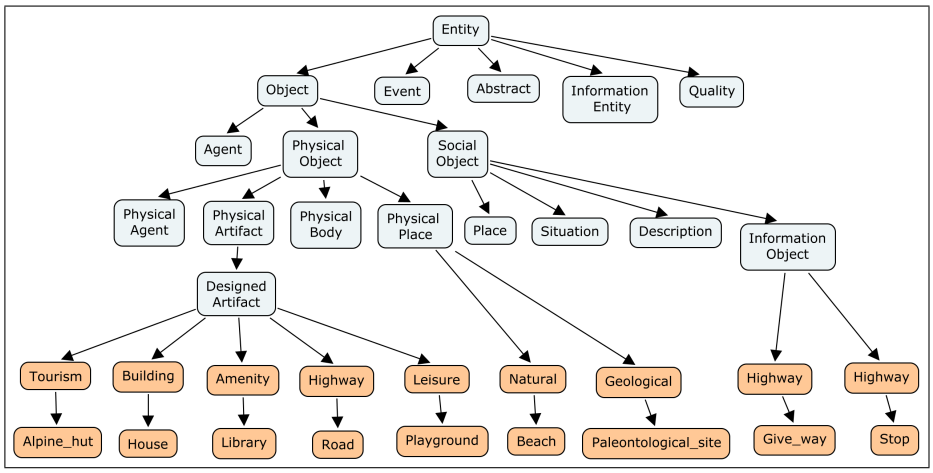


Fig. 1. Tags aligned to DOLCE+ DnS Ultralite

show agreement or disagreement between users i.e. for the same tag what kind of answers users provide and to which alignment it leads. Results can then be evaluated with the aid of domain experts.

By introducing the "question game" for the alignment process, a solution is found that allows users to keep the freedom of choice for their tags while at the same time it enables the anchoring of them in the top level ontology.

5 Conclusion and Future Work

In this paper we have shown a way to bridge the gap between top-down ontological and bottom-up crowdsourcing practices. OpenStreetMap is chosen as a representative, widely used example of VGI. Although the freedom of assigning tags to OpenStreetMap is very appealing and encouraging for users to contribute their geodata, it humbles information search and retrieval. The community tries to stabilize a common agreement on tags and an appropriate way that they be used; however a proper order is missing. As a result, implicit knowledge (i.e. inherent characteristics between classes and subclasses) cannot be unfolded.

The need for ordering the OpenStreetMap tags and constraining their meaning, was fulfilled with the alignment of the tags to the DOLCE Ultralite top level ontology. DOLCE Ultralite has a cognitive and linguistic orientation and was therefore preferable to other top level ontologies. A bottom-up method is used for the alignment process preserving the open and dynamic character of OpenStreetMap.

With the aid of a "question game", users are guided to align their tags to DOLCE concepts. Keys and values of OpenStreetMap are translated into classes and subclasses respectively and then anchored. Users are neither confronted with

the concepts of DOLCE Ultralite nor the DOLCE Ultralite hierarchy so that the simple and common tagging procedure is maintained.

Opportunities for future work comprise the use of sophisticated reasoning mechanisms to derive the implicit knowledge from the alignment result. Especially the analysis of inconsistencies between tags would provide evidence about which geographic entities are perceived and used differently among the users. For instance, if the alignment results show that the same tag is anchored in different DOLCE Ultralite concepts it can be inferred that the geographic entity it describes, is conceptualized in heterogeneous ways by each user.

With additional analysis of these findings, new knowledge could be derived. On a higher generalization level, this would be a way to derive a better understanding of how users conceptualize geographic entities providing insights to spatial cognition.

Given the fact that OpenStreetMap is a multilingual project one research question to be further investigated, would be whether tags in different languages are aligned to the same DOLCE Ultralite concept or not. Such an analysis could assist in investigating if there are differences in the conceptualization of the same POI in different cultures.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme - Marie Curie Actions, Initial Training Network GEOCROWD under grant agreement n FP7-PEOPLE-2010-ITN-264994.

References

1. S. Angeletou, M. Sabou, L. Specia, and E. Motta. Bridging the gap between folksonomies and the semantic web: An experience report. In *Workshop: Bridging the Gap between Semantic Web and Web 2.0, European Semantic Web Conference*, page 93, 2007.
2. S. Auer, J. Lehmann, and S. Hellmann. LinkedGeoData - adding a spatial dimension to the web of data. In *Proc. of 8th International Semantic Web Conference (ISWC)*, 2009.
3. A. Ballatore and M. Bertolotto. Semantically enriching vgi in support of implicit feedback analysis. In *Proceedings of the 10th international conference on Web and wireless geographical information systems*, W2GIS'11, pages 78–93, Berlin, Heidelberg, 2011. Springer-Verlag.
4. G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, pages 22–26, 2006.
5. J. Chamberlain, M. Poesio, and U. Kruschwitz. Phrase detectives: A web-based collaborative annotation game. In *Proceedings of the International Conference on Semantic Systems (I-Semantics 08), Graz*, 2008.

6. M. Codescu, G. Horsinka, O. Kutz, T. Mossakowski, and R. Rau. Do-roam: activity-oriented search and navigation with openstreetmap. *GeoSpatial Semantics*, pages 88–107, 2011.
7. M. Codescu, G. Horsinka, O. Kutz, T. Mossakowski, and R. Rau. Osmonto-an ontology of openstreetmap tags. *State of the map Europe (SOTM-EU) 2011*, 2011.
8. S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.
9. S. Cosat. Openstreetmap-the best map. `http://opengeodata.org/openstreetmap-the-best-map`, 2010. Last accessed: 27.07.2012.
10. A. Gangemi, N. Guarino, A. Masolo, C.and Oltramari, and L. Schneider. Sweetening ontologies with dolce. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 166–181, London, UK, UK, 2002. Springer-Verlag.
11. B.M. Good and A.I. Su. Games with a scientific purpose. *Genome Biology*, 12(12):135, 2011.
12. M.F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.
13. M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.
14. M. Kavouras and M. Kokla. *Theories of Geographic Concepts: Ontological Approaches to Semantic Integration*. Taylor & Francis, 2007.
15. C.S.G. Khoo and J.C. Na. Semantic relations in information science. *Annual Review of Information Science and Technology*, 40:157, 2006.
16. E. Klien. Deliverable D3.1 Ontologies in the swing application requirement specification. Technical report, FP6-26514 Project: Semantic Web-Service Interoperability for Geospatial Decision Making, March 2007.
17. D. Laniado, D. Eynard, and M. Colombetti. Using wordnet to turn a folksonomy into a hierarchy of concepts. In *Semantic Web Application and Perspectives - Fourth Italian Semantic Web Workshop, Bari Italy*, pages 192–201, December 2007.
18. K. Latif, E. Weippl, and A. M. Tjoa. Question driven semantics interpretation for collaborative knowledge engineering and ontology reuse. In *IRI*, pages 170–176. IEEE Systems, Man, and Cybernetics Society, 2007.
19. H. Lin and J. Davis. Computational and crowdsourcing methods for extracting ontological structure from folksonomy. In *Proceedings of the 7th international conference on The Semantic Web: research and Applications - Volume Part II*, ESWC'10, pages 472–477, Berlin, Heidelberg, 2010. Springer-Verlag.
20. T.W. Malone, R. Laubacher, and C. Dellarocas. Harnessing crowds: Mapping the genome of collective intelligence. Research Paper No. 4732-09, MIT, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA, February 2009.
21. C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. WonderWeb deliverable D18 ontology library (final). Technical report, IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web, 2003.
22. P. Mooney and P. Corcoran. Annotating spatial features in open-streetmap. *Proceedings of the 19th annual gis research uk (gisruk), Portsmouth, England*, 2011.
23. P. Mooney and P. Corcoran. Characteristics of heavily edited objects in openstreetmap. *Future Internet*, 4(1):285–305, 2012.
24. A. Passant. Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs. In *International Conference on Weblogs and Social Media, Boulder, Colorado*. ICWSM, March 2007.

25. N. Prestopnik and K. Crowston. Exploring collective intelligence games with design science: A citizen science design case. In *ACM Group Conference*, Sanibel Island, FL, October 2012. ACM.
26. S. Scheider, C. Kessler, J. Ortmann, A. Devaraju, J. Trame, T. Kauppinen, and W. Kuhn. Semantic referencing of geosensor data and volunteered geographic information. In Naveen Ashish and Amit P. Sheth, editors, *Geospatial Semantics and the Semantic Web*, volume 12 of *Semantic Web And Beyond Computing for Human Experience*, pages 27–59. Springer, 2011.
27. K. Siorpaes and M. Hepp. Ontogame: towards overcoming the incentive bottleneck in ontology building. In *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, pages 1222–1232. Springer, 2007.
28. K. Siorpaes and M. Hepp. Games with a purpose for the semantic web. 23:50–60, 2008.
29. L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *Proc. of the European Semantic Web Conference (ESWC2007)*, volume 4519 of *LNCS*, pages 624–639, Berlin Heidelberg, Germany, July 2007. Springer-Verlag.
30. S. Thaler, E. Simperl, and K. Siorpaes. Spotthelink: A game for ontology alignment. In Ronald Maier, editor, *Proceedings of the 6th Conference for Professional Knowledge Management WM*, volume 182 of *LNI*, pages 246–253. GI, 2011.
31. S. Thaler, K. Siorpaes, E. Simperl, and C. Hofer. A survey on games for knowledge acquisition. Technical report, Tech. Rep. STI TR 2011-05-01, Semantic Technology Institute, 2011.
32. D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet. A Game-Based Approach for Collecting Semantic Annotations of Music. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, September 2007.
33. T. Vander Wal. Folksonomy coinage and definition. `http://vanderwal.net/folksonomy.html`, 2007. Last accessed: 27.07.2012.
34. L. von Ahn. Games with a Purpose. *Computer*, 39(6):92–94, June 2006.
35. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

Querying Linked Geospatial Data with Incomplete Information

C. Nikolaou and M. Koubarakis

Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece
charnik@di.uoa.gr

Abstract. Linked geospatial data has recently received attention, as researchers and practitioners have started tapping the wealth of geospatial information available on the Web. Incomplete geospatial information, although appearing often in the applications captured by such datasets, is not represented and queried properly due to the lack of appropriate data models and query languages. We discuss our recent work on the model RDF$^{\mathrm{i}}$, an extension of RDF with the ability to represent property values that exist, but are unknown or partially known, using constraints, and an extension of the query language SPARQL with qualitative and quantitative geospatial querying capabilities. We demonstrate the usefulness of RDF$^{\mathrm{i}}$ in geospatial Semantic Web applications by giving examples and comparing the modeling capabilities of RDF$^{\mathrm{i}}$ with the ones of related Semantic Web systems.

Keywords: linked geospatial data, incomplete information, RDF

1 Introduction

Linked data is a new research area which studies how one can make RDF data available on the Web, and interconnect it with other data with the aim of increasing its value for everybody [4]. The resulting "Web of data" has recently started being populated with geospatial data. A representative example of such efforts is LinkedGeoData[1] where OpenStreetMap data is made available as RDF and queried using the declarative query language SPARQL [2]. With the recent emphasis on open government data, some of it encoded already in RDF[2], portals such as LinkedGeoData demonstrate that the development of useful Web applications might be just a few SPARQL queries away. The recent paper [9] by our group addresses many research topics and relevant questions that deserve the attention of researchers in the area of linked geospatial data.

In the context of the research agenda presented in [9], we have developed stSPARQL [17], an extension of the query language SPARQL for querying linked

[1] http://linkedgeodata.org/
[2] http://data.gov.uk/linked-data/

geospatial data. The geospatial component of stSPARQL has been fully implemented in our open source system Strabon[3] which also supports GeoSPARQL, the recent proposed standard by OGC (Open Geospatial Consortium) for querying geospatial data expressed in RDF. Strabon is currently being used to query linked data describing sensors in the context of project SemsorGrid4Env[4] [16] and linked earth observation (EO) data in the context of project TELEIOS[5] [12].

A significant aspect of querying linked geospatial data that has not been addressed yet is querying linked geospatial data with incomplete information [11]. Incomplete information, although appearing often in applications captured by such datasets, is not represented or queried properly due to the lack of appropriate data models and query languages. For example, a wildfire monitoring and management application, developed by us in TELEIOS, requires the integration of multiple, heterogeneous data sources, some of them available on the Web, with data of varying quality and varying temporal and spatial scales. As a result, incomplete information needs to be represented in stRDF and queried by stSPARQL.

In this paper we address the problem of representing and querying *incomplete geospatial information* in RDF using the RDF$^{\text{i}}$ framework that we have recently developed in [19]. RDF$^{\text{i}}$ is a framework that extends RDF with the ability to represent property values that exist, but are unknown or partially known, using constraints. RDF$^{\text{i}}$ is a general framework for the representation of incomplete information of this kind and it can be employed in various application domains, such as temporal and spatial. In this paper, we concentrate on the spatial domain only and demonstrate the modeling capabilities of RDF$^{\text{i}}$ and the querying capabilities of our extension of SPARQL which is based on stSPARQL.

The organization of the paper is as follows. Section 2 introduces the RDF$^{\text{i}}$ framework. Section 3 describes the kinds of linked geospatial data that we need to represent in the wildfire monitoring application of TELEIOS. Then, Section 4 demonstrates the RDF$^{\text{i}}$ framework giving examples motivated from that application of TELEIOS. Finally, Section 5 compares the expressive power of RDF$^{\text{i}}$ with related semantic web systems, while Section 6 concludes our work.

The paper is mostly informal and uses examples from the wildfire monitoring application of TELEIOS. Even in the places where the paper becomes formal, we do not give any detailed technical results for which the interested reader is directed to [13, 14, 19] and the survey paper [9].

2 The RDF$^{\text{i}}$ framework

The RDF$^{\text{i}}$ framework developed by us in [19] (where "i" stands for "incomplete") is an extension of the RDF framework addressing an important kind of incomplete information that has so far been ignored in the context of RDF;

[3] `http://www.strabon.di.uoa.gr/`
[4] `http://www.semsorgrid4env.eu/`
[5] `http://www.earthobservatory.eu/`

representation of values that *exist but are unknown or partially known*. RDF$^{\text{i}}$ extends RDF with the ability to define a new kind of literals for each datatype. These literals are called *e-literals* ("e" comes from the word "existential") and can be used to represent values of properties that *exist but are unknown or partially known*. Such information is abundant in recent applications where RDF is being used (e.g., sensor networks, the modeling of geospatial information, etc.). In RDF$^{\text{i}}$, e-literals are allowed to appear only in the object position of triples.

Previous research on incomplete information in databases and knowledge representation has shown that in many applications, having the ability to state *constraints* about values that are partially known is a very desirable feature and leads to the development of very expressive formalisms [5, 8]. In the spirit of this tradition, RDF$^{\text{i}}$ allows partial information regarding property values represented by e-literals to be expressed by a quantifier-free formula of a first-order *constraint language* $\mathcal{L}$. Thus, RDF$^{\text{i}}$ extends the concept of an RDF graph to the concept of an RDF$^{\text{i}}$ *database* which is a pair (G, ϕ) where G is an RDF graph possibly containing triples with e-literals in their object positions, and ϕ is a quantifier-free formula of $\mathcal{L}$.

The semantics for RDF$^{\text{i}}$ databases and SPARQL query evaluation has been defined following ideas from the incomplete information literature [5, 6]. The semantics defines the set of possible RDF graphs corresponding to an RDF$^{\text{i}}$ database and the fundamental concept of certain answer for SPARQL query evaluation over an RDF$^{\text{i}}$ database.

The well-known concept of *representation system* from the seminal paper of [6] has been transferred to the case of RDF$^{\text{i}}$. It has been shown in [19] that CONSTRUCT queries without blank nodes in their templates and using only the operators AND, UNION, and FILTER or the restricted fragment of graph patterns corresponding to the well-designed patterns of [1] can be used to define a representation system for RDF$^{\text{i}}$. Last, [19] defines the fundamental concept of certain answer to SPARQL queries over RDF$^{\text{i}}$ databases and presents an algorithm for its computation.

3 Linked geospatial data in the wildfire monitoring application of TELEIOS

The wildfire monitoring application of TELEIOS concentrates on the development of solutions for real time hotspot and active fire front detection, and burnt area mapping. Technological solutions to both of these cases require integration of multiple, heterogeneous data sources with data of varying quality and varying temporal and spatial scales. Some of the data sources are streams (e.g., streams of EO images) while others are static geo-information layers (e.g., land use/land cover maps) providing additional evidence on the underlying characteristics of the affected area.

In what follows, we briefly describe some of the datasets used by the National Observatory of Athens (NOA) that is leading the wildfire monitoring application of TELEIOS.

Hotspot maps. NOA operates a MSG/SEVIRI[6] acquisition station and receives raw satellite images every 15 minutes. These images are processed using image processing algorithms to detect the existence of hotspots. Information related to hotspots is stored in ESRI shapefiles and KML files. These files hold information about the date and time of image acquisition, cartographic X, Y coordinates of detected fire locations, the level of reliability in the observations, the fire radiative power assessed, and the observed fire area. NOA receives similar hotspot shapefiles covering the geographical area of Greece from the European project SAFER (Services and Applications for Emergency Response).

Burnt area maps. From project SAFER, NOA also receives ready-to-use accumulated burnt area mapping products in polygon format, projected to the EGSA87 reference system[7]. These products are derived daily using the MODIS satellite and cover the entire Greek territory. The data formats are ESRI shapefiles and KML files with information relating to date and time of image acquisition, and the mapped fire area.

Corine Land Cover data. The Corine Land Cover project is an activity of the European Environment Agency which is collecting data regarding land cover (e.g., farmland, forest) of European countries. The Corine Land Cover nomenclature uses a hierarchical scheme with three levels to describe land cover:

- The first level consists of five items and indicates the major categories of land cover on the planet, e.g., forests and semi-natural areas.
- The second level consists of fifteen items and is intended for use on scales of 1:500,000 and 1:1,000,000 identifying more specific types of land cover, e.g., open spaces with little or no vegetation.
- The third level consists of forty-four items and is intended for use on a scale of 1:100,000, narrowing down the land use to a very specific geographic characterization, e.g., burnt areas.

The land cover of Greece is available as an ESRI shapefile that is based on the Corine Land Cover nomenclature.

Coastline geometry of Greece. An ESRI shapefile that describes the geometry of the coastline of Greece is available.

In [15, 17] we discuss in great detail how we can query linked geospatial data such as the above using the model stRDF and the query language stSPARQL. In this work we concentrate on the representation and querying of linked geospatial data with incomplete information. This is presented in the following section by giving examples motivated from the wildfire monitoring application of TELEIOS.

[6] MSG refers to Meteosat Second Generation satellites, and SEVIRI is the instrument which is responsible for taking infrared images of the earth.

[7] EGSA87 is a 2-dimensional projected coordinate reference system that describes the area of Greece.

4 Incomplete geospatial information in the wildfire monitoring application of TELEIOS

This section motivates our approach towards extending RDF with the ability to represent and query incomplete information.

As mentioned in Section 3, NOA receives satellite images for the entire Greek fire season on a 15-minute basis from the SEVIRI infrared imager of a Meteosat Second Generation satellite. After the images are processed for georeferencing, they are analyzed by specialized image processing software to detect hotspots (i.e., regions of the image corresponding to geographic regions that are probably on fire). Processing of images results in the generation of shapefiles representing hotspots as point-vectors.

The following is a list of triples (namespaces are omitted) that gives an example of the kind of representation that is currently used by NOA for representing these hotspots and making them available as linked data to relevant public authorities.

```
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occuredIn region1 .
region1 hasGeometry "x = 24.825668 ∧ y = 35.310643"^^SemiLinearPointSet .
```

The above list of triples is a graph in the model stRDF of [10] which extends RDF with the ability to represent geometries over $\mathbb{Q}^k$ that change over time following the paradigm of constraint databases [7]. In stRDF, geometries and valid times of triples are expressed using Boolean combinations of linear constraints that are given as literals of type `SemiLinearPointSet` defined in [10]. *Semi-linear point sets* are the subsets of $\mathbb{Q}^k$ defined by Boolean combinations of linear constraints. The above graph represents *definite* information; it states that there is a hotspot (`hotspot1`) and that the corresponding fire (`fire1`) takes place at the point $(24.825668, 35.310643) \in \mathbb{Q}^2$.

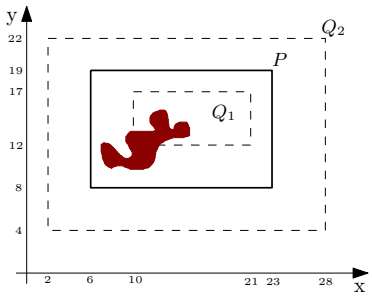


Fig. 1. Rectangles mentioned in the examples

In practice, due to the technical weaknesses of the instruments attached to satellites and inherent distortions of the algorithms applied on satellite images for knowledge extraction, the extracted spatial information can only be indefinite. For example, the SEVIRI imager has medium resolution and therefore each image pixel representing a hotspot corresponds to a 3km by 3km rectangle in geographic space. Accordingly, NOA represents hotspots as points in geographic space using the center of the corresponding rectangle.

In this case, another useful representation of the real world situation that corresponds to a hotspot would be to state that there is a geographic region with unknown exact coordinates where a fire is taking place, and that region is included in a known 3km by 3km rectangle. This real world situation can be represented by an RDF$^{\text{i}}$ database as shown in the following example.

Example 1. The following is an RDF$^{\text{i}}$ database encoding information about a detected hotspot.

```
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occuredIn _R1 .
```

$$\_\text{R1 NTPP "x} \geq 6 \wedge \text{x} \leq 23 \wedge \text{y} \geq 8 \wedge \text{y} \leq 19\text{"}$$

Fire `fire1` (red area of Figure 1) is asserted to have taken place inside region _R1. _R1 is an e-literal of datatype `SemiLinearPointSet` and is asserted to be inside the rectangle formed by the points $(6, 8)$ and $(23, 19)$ (rectangle P of Figure 1)[8]. This is stated with a constraint expressed in the language PCL (Polygon Constraint Language), a first-order constraint language that allows us to represent topological properties for polygons. NTPP is the "non-tangential-proper-part" relation of RCC-8 [24]. In general, constraints in PCL can be used to express qualitative and quantitative spatial information about regions in $\mathbb{Q}^2$.

The example shows that e-literals are like existentially quantified variables in first-order logic or Skolem constants. E-literals can be used to represent values of properties that *exist* but are *unknown* or *partially known* (e.g., by constraining the value of an e-literal).

RDF$^{\text{i}}$ databases like the one of Example 1 consist of two parts: a graph (i.e., a set of triples) and a *global constraint*. Global constraints can in general be quantifier-free formulae of some first-order constraint language. RDF$^{\text{i}}$ databases are syntactic devices for the representation of incomplete information. An RDF$^{\text{i}}$ database is semantically equivalent to a set of possible RDF graphs that represent all the possible ways the domain of application could have been according to our incomplete information. One can find all the possible RDF graphs represented by an RDF$^{\text{i}}$ database as follows: *a*) find an assignment to e-literals that satisfies the global constraint and *b*) substitute these values for the e-literals in the RDF$^{\text{i}}$ database.

[8] For sake of readability of the examples, we chose to use small, integer numbers instead of real geographic coordinates.

For example, the RDF graph shown below is one of the possible RDF graphs corresponding to the RDF$^{\text{i}}$ database of Example 1.

```
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occuredIn "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17" .
```

RDF$^{\text{i}}$ databases can be queried using the well-known query language SPARQL. Example 2 below demonstrates a query in SPARQL.

Example 2. Let us consider the query "Find all fires that have occurred in a region which is a non-tangential proper part of rectangle Q_1 of Figure 1" over the database of Example 1. In the extension of SPARQL we consider, this query can be expressed as follows:

```
SELECT ?F
WHERE  {
            ?F type Fire .
            ?F occuredIn ?R .
            FILTER ( NTPP(?R, "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17") )
}
```

The version of SPARQL we consider extends FILTER expressions of standard SPARQL [23] allowing also expressions of a first-order constraint language, such as PCL, for constraining the values of spatial variables. These expressions have a functional-like syntax and are interpreted in the underlying first-order language. For example, the global constraint of the RDF$^{\text{i}}$ database of Example 1 would be specified in a FILTER expression as

```
NTPP(?R, "x ≥ 6 ∧ x ≤ 23 ∧ y ≥ 8 ∧ y ≤ 19")
```

to constrain the value of the spatial variable `?R`.

What is the answer to the query of Example 2? If we examine the database of Example 1 (Figure 1), we can see that the answer should be *conditional* [6]. We cannot say for sure whether `fire1` satisfies the requirements of the query because the information in the database is indefinite (the exact geometry of `_R1` is not known). Fire `fire1` qualifies only in the possible graphs where `_R1` is a non-tangential proper part of the rectangle mentioned in the query. For every object that qualifies as an answer, the query answering procedure should also provide a *condition* characterizing this set of possible graphs. Following the ideas of conditional tables from [6], this answer can be represented by the following set of *conditional mappings* (see [19] for a formal definition):

?F	Condition
`fire1`	_R1 NTPP "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17"

Conditional mappings are different from standard SPARQL mappings [22] in the sense that they map variables to constants only if a condition holds. Thus,

they are reminiscent of conditional tuples in the conditional table model of [5]. In RDF^{i}, the basic concept of triple is also defined to be conditional.

Example 3. If we wanted to have an RDF^{i} database as the answer to a query like the one of Example 2, then we would have queried the RDF^{i} database of Example 1 using the CONSTRUCT query form of SPARQL as follows:

```
CONSTRUCT { ?F type Fire }
WHERE     {
              ?F type Fire .
              ?F occuredIn ?R .
              FILTER ( NTPP(?R, "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17") )
}
```

The answer to this query would be an RDF^{i} database containing *conditional triples* adhering to the query template (i.e., {`?F type Fire`}). The template is instantiated for each conditional mapping from the evaluation of the graph pattern of the query, and the resulting triple together with the condition of the mapping form a conditional triple in the resulting database. Therefore, the answer to query of Example 3 consists of the following conditional triple:

`fire1 type Fire [`_R1 NTPP "x ≥ 10 ∧ x ≤ 21 ∧ y ≥ 12 ∧ y ≤ 17"] .

The e-literals in the above answer (i.e., _R1) are implicitly constrained by the global constraint of the original database, i.e., constraint

_R1 NTPP "x ≥ 6 ∧ x ≤ 23 ∧ y ≥ 8 ∧ y ≤ 19".

In some cases the user might know that the information in the database is incomplete. Thus, she might wish to find all values that *certainly* satisfy some qualification. This is the well-known notion of certain answer in the incomplete databases literature [5] and it is demonstrated in the following example.

Example 4. Let us consider the query of Example 3 again and rephrase it to "Find fires that have *certainly* occurred in a region which is a non-tangential proper part of rectangle Q_2 of Figure 1". In the version of SPARQL we consider, this query would be expressed as follows:

```
CERTAIN CONSTRUCT { ?F type Fire }
WHERE  {
            ?F type Fire .
            ?F occuredIn ?R .
            FILTER ( NTPP(?R, "x ≥ 2 ∧ x ≤ 28 ∧ y ≥ 4 ∧ y ≤ 22") )
}
```

Inspecting Figure 1, it is obvious that `fire1` satisfies the query unconditionally. Hence, the certain answer contains the following RDF triple

`fire1 type Fire .`

In contrast to Example 3 where the answer to the CONSTRUCT query is an RDF$^{\text{i}}$ database, the answer to a CONSTRUCT query with a CERTAIN operator, like the one of Example 4 above, is an RDF graph. This is anticipated since a certain answer can not contain conditional information.

5 Expressive power of RDF$^{\text{i}}$: An informal comparison

In this paper we gave examples of the use of RDF$^{\text{i}}$ in geospatial applications. Thus, it would be interesting to compare the expressive power that RDF$^{\text{i}}$ gives us to other recent works that use Semantic Web data models and languages for geospatial applications.

When equipped with a constraint language like PCL (or TCL[9]) [19], RDF$^{\text{i}}$ goes beyond the proposals of [10, 17] and [20] that cannot express incomplete geospatial information. Incomplete geospatial information as it is studied in this paper can also be expressed in spatial description logics [18, 21]. For efficiency reasons, spatial DL reasoners such as RacerPro[10] and PelletSpatial[11] have opted for separating spatial relations from standard DL axioms as we have done by separating graphs and constraints. Since RDF graphs can be seen as DL ABoxes with atomic concepts only, all the results of this paper can be trivially transferred to the relevant subsets of spatial DLs and their reasoners.

In the following we concentrate on the reasoner PelletSpatial since it is a more recent proposal than RacerPro and discuss how RDF$^{\text{i}}$ is related to the recently proposed Semantic Web technologies of [3, 26].

PelletSpatial [25] is a hybrid spatial reasoner that provides RCC-8 and OWL 2 reasoning and querying capabilities. In PelletSpatial, spatial relations are separated from OWL 2 relations providing a hybrid reasoner for both spatial and thematic data. Spatial relations are managed as an RCC-8 constraint network. Conjunctive query answering in PelletSpatial requires two phases: *a*) evaluating spatial query atoms over the constraint network by employing a path-consistency algorithm, and *b*) further constraining the set of bindings such that the non-spatial query atoms are satisfied.

Compared to the RDF$^{\text{i}}$ framework, PelletSpatial corresponds to RDF$^{\text{i}}$ databases with a conjunction of TCL-constraints as a global constraint. Compared to our extension of SPARQL, the query language of PelletSpatial computes certain answers for SPARQL queries using only the operators AND and FILTER with conjunctions of TCL-constraints allowed as expressions in FILTER graph patterns. The representational and querying power of RDF$^{\text{i}}$ when $\mathcal{L}$ is PCL is greater than the one of PelletSpatial since PCL is a language more expressive than TCL. However, PelletSpatial offers OWL representation and reasoning that is not offered by RDF$^{\text{i}}$.

[9] TCL is like PCL but without constants, that is, TCL can express topological constraints only between variables.

[10] `http://www.racer-systems.com/`

[11] `http://clarkparsia.com/pellet/spatial/`

A more general and formal approach to modeling spatial information is [26] that proposes an abstracted graph-based data model and query language with which any subset of first-order predicate logic (FOPL) (e.g., modal, description logic) can be associated. For the case of spatial information, a substrate can play the role of a geometric substrate, called SBox. SBox deals with spatial datatypes (e.g., polygons) the geometry of which can be described using an appropriate FOPL, inheriting also its formal semantics for satisfiability, entailment, etc. The authors investigate four options for representing and querying spatial information: use (i) an ABox, (ii) a map substrate, (iii) a spatial ABox, (iv) an ABox and RCC substrate.

Finally, [3] proposes SOWL, an extension of OWL, to represent spatial qualitative and quantitative information employing the RCC-8 topological relations, cardinal direction relations, and distance relations. To reason about spatial relations, a set of SWRL rules are implemented in the Pellet reasoner.

6 Conclusions

This work stressed the inability of semantic web data models and query languages to manage linked geospatial data with incomplete information. Motivated by a real application in which representation and querying of incomplete information is inherent, it demonstrated through the use of many examples how the RDF$^{\text{i}}$ framework and an extension of the query language SPARQL [19] can be employed for active fire front detection and burnt area mapping in the context of the EU project TELEIOS.

Acknowledgments

This work has been funded by the FP7 project TELEIOS (257662).

References

1. M. Arenas and J. Pérez. Querying semantic web data with SPARQL. In *PODS*, pages 305–316, 2011.
2. S. Auer, J. Lehmann, and S. Hellmann. Linkedgeodata: Adding a spatial dimension to the web of data. In *ISWC'09*, pages 731–746, 2009.
3. S. Batsakis and E. Petrakis. SOWL: spatio-temporal representation, reasoning and querying over the semantic web. In *I-SEMANTICS*, 2010.
4. C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
5. G. Grahne. *The Problem of Incomplete Information in Relational Databases*, volume 554 of *LNCS*. Springer Verlag, 1991.
6. T. Imielinski and W. Lipski. Incomplete Information in Relational Databases. *JACM*, 31(4):761–791, 1984.
7. P. C. Kanellakis, G. M. Kuper, and P. Z. Revesz. Constraint Query Languages. In *PODS*, pages 299–313, 1990.

8. M. Koubarakis. Database models for infinite and indefinite temporal information. *Inf. Syst.*, 19(2):141–173, 1994.
9. M. Koubarakis, M. Karpathiotakis, K. Kyzirakos, C. Nikolaou, and M. Sioutis. Data Models and Query Languages for Linked Geospatial Data. In T. Eiter and T. Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering*, volume 7487 of *Lecture Notes in Computer Science*, pages 290–328. Springer Berlin / Heidelberg, 2012.
10. M. Koubarakis and K. Kyzirakos. Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In *ESWC*, 2010.
11. M. Koubarakis, K. Kyzirakos, M. Karpathiotakis, C. Nikolaou, M. Sioutis, S. Vassos, D. Michail, T. t. Herekakis, C. Kontoes, and I. Papoutsis. Challenges for Qualitative Spatial Reasoning in Linked Geospatial Data. In *Proceedings of IJCAI 2011 Workshop on Benchmarks and Applications of Spatial Reasoning*, 2011.
12. M. Koubarakis, K. Kyzirakos, M. Karpathiotakis, C. Nikolaou, S. Vassos, G. Garbis, M. Sioutis, K. Bereta, D. Michail, C. Kontoes, I. Papoutsis, T. Herekakis, S. Manegold, M. Kersten, M. Ivanova, H. Pirk, Y. Zhang, M. Datcu, G. Schwarz, C. Dumitru, D. E. Molina, K. Molch, U. D. Giammatteo, M. Sagona, S. Perelli, T. Reitz, E. Klien, and R. Gregor. TELEIOS: A Database-Powered Virtual Earth Observatory. *PVLDB*, 5(12):2010–2013, 2012.
13. M. Koubarakis, C. Nikolaou, and V. Fisikopoulos. Theoretical results on query processing for RDF/SPARQL with time and space. Del. 2.3, TELEIOS, 2011.
14. M. Koubarakis et al. A data model and query language for an extension of RDF with time and space. Del. 2.1, TELEIOS project, 2011.
15. K. Kyzirakos, M. Karpathiotakis, G. Garbis, C. Nikolaou, K. Bereta, M. Sioutis, I. Papoutsis, T. Herekakis, D. Michail, M. Koubarakis, and H. Kontoes. Real Time Fire Monitoring Using Semantic Web and Linked Data Technologies. In *ISWC'12*, 2012.
16. K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. Developing Registries for the Semantic Sensor Web using stRDF and stSPARQL. In *SSN*, volume 668, 2010.
17. K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. Strabon: A Semantic Geospatial DBMS. In *ISWC'12*, 2012.
18. C. Lutz and M. Miličić. A tableau algorithm for description logics with concrete domains and general tboxes. *J. Autom. Reason.*, 38:227–259, April 2007.
19. C. Nikolaou and M. Koubarakis. Incomplete Information in the Semantic Web. 2012 (submitted to a conference).
20. Open Geospatial Consortium Inc. GeoSPARQL - A geographic query language for RDF data. OGC, 2010.
21. Ö. L. Özçep and R. Möller. Combining DL-Lite with Spatial Calculi for Feasible Geo-thematic Query Answering. In *Description Logics*, 2012.
22. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3):1–45, 2009.
23. E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Recommendation 15 Jan. 2008.
24. D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.
25. M. Stocker and E. Sirin. PelletSpatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine. In *OWLED*, 2009.
26. M. Wessel and R. Moller. Flexible software architectures for ontology-based information systems. *JAL*, 7(1):75–99, 2009.

Ontology-based Route Planning for OpenStreetMap

Mihai Codescu[1], Daniel Couto Vale[2], Oliver Kutz[2], and Till Mossakowski[1,2]

[1] DFKI GmbH Bremen
[2] SFB/TR 8 Spatial Cognition, University of Bremen, Germany

Abstract. We develop a web service for route finding in OpenStreetMap (OSM) following an activity-centred approach: the aim is not only to assist the user in travelling from A to B, but also to perform a series of specified activities along the way. This is particularly important e.g. for electric mobility, where activities can take place while the battery of the car is recharging. For specifying activities, our tool uses an ontology of spatially-located activities. This ontology then needs to be related to OSM which provides semantic metadata in form of tags. We organised the tags into another ontology (which is then connected to the first one via an ontology mapping), providing thus not only better reference for the OSM community, but also allowing to enrich the ontological semantics of the tags, to deal with their evolving nature and to extract implicit information using ontology-based data access. Moreover, we report first results on an ongoing query corpus experiment involving the OSM community targeted at improving the automated understanding of free text input for certain route finding tasks.

1 Introduction

We develop a web-based system DO-ROAM[3] for activity-oriented route planning. In our system, geolocations can be specified in different ways for finding a route. While traditional route planning works with identifiable locations referred to by names or coordinates, DO-ROAM allows specification of kinds of activities and facilities, whose identities and locations are only determined later on. It also includes a simple temporal planning component.

Several aspects of a massive data-intensive system such as DO-ROAM must be tackled. On the content aspect, it is unfeasible to produce the necessary amount of data in a centralised way. For this reason DO-ROAM draws on data from OpenStreetMap, which provides not only entity coordinates, but also a fair amount of metadata, like their names, opening hours, activities, URLs and the like. Since metadata obtained as volunteered geographical information (VGI) through collaborative and community-based efforts evolve in a bottom-up way contain a lot of noise (typos, redundancies, etc.) and are subject to constant change, we intend to make the access to information such as the OpenStreetMap data more structured and stable through the application of ontologies. Our goal is to extract an ontology from VGI in an automated way.

On the usability aspect of our system, offering an intuitive user interface makes it more likely that a web-based tool appeals to a broad user base. Route planning is

[3] A prototype is freely available at www.do-roam.org; DO-ROAM stands for *Data and Ontology driven Route-finding Of Activity-oriented Mobility*.

an intrinsically complex activity. In the interaction model of DO-ROAM, one can, for instance, browse the map laterally, zoom in and out, localise oneself, and, most importantly, specify either a location or a route composed of an origin, a destination and possibly intermediary activities. For this last task, we offer two specification methods, a guided one through drop down menus and an unguided one through a text field. Very central to the intuitiveness of this interface is the correct analysis of most direction queries frequently typed in the text field according to the linguistic conventions of each covered locale.

This work extends [6] in various ways. The motivating use cases in Sect. 2 are new. The architecture of the tool, presented in Sect. 3, has been updated from [6] to include the route planning component. The ontology of tags, OSMonto, is structured in a cleaner way (using roles, see Sect. 4), leading to a new version of the ontology mapping that links OSMonto to the ontology of activities, designed in [6] (Sect. 5). Some details regarding generation and maintenance of OSMonto can be found in [6]. Finally, we present the results of an experiment conducted with German native speakers that will support an automatic analysis of direction queries for the German-speaking regions of Europe (Sect. 6).

2 Motivating Use Cases

In the following, we describe how our system can be used in two complex use cases to specify activities and plan a route through locations where one can do them. In creating use cases, one of our concerns was electric mobility because of the rather limited reach of electric cars and relatively long charging times. The maps are provided by OpenStreetMap, but the approach is flexible enough to include data from other sources. [1] presents several navigation scenarios based on a GIS ontology; this work inspired the use cases presented here. We now describe how DO-ROAM supports use cases 1 and 2.

Fig. 1. Use case 1: activity selection.

Use Case 1: Betty is a tourist and wants to find out which activities are in reach by foot from the nearest charging station. She also knows that she will be getting hungry soon and thus looks for all restaurant which will be open the next 2 hours.

Use Case 2: Maria wants to visit a friend in Bremen. On her way from the train station to her friend's flat she would like to go shopping and visit a bank and a post office. She needs a system generating a route containing all these places.

The screenshot in Fig. 1 illustrates how to use the interface to find all restaurants open during the next 2 hours. In use case 2, Maria should first select a start and end point of a route. Afterwards she should add places for shopping, a bank and a post office. A routing algorithm must then be selected (in our case OSRM) and then she clicks “Calculate route” which generates the route shown in Fig. 2. The system also supports resetting the route at any given time.

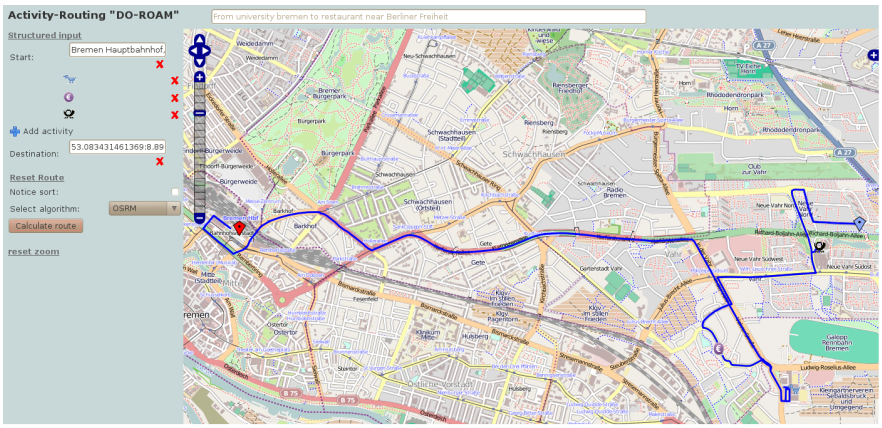


Fig. 2. Use case 2: route planning

3 Architecture of DO-ROAM

DO-ROAM is a web application intended to assist spatio-temporal planning of activities and routes. At the moment, a prototype is available at http://do-roam.org. It is implemented in Ruby on Rails and its architecture is shown in the Fig. 3. A previous version of DO-ROAM, described in [6], was developed on top of the OSM rails port. We have chosen to build our system as a new, clean Rails project to gain simplicity. Our work has been inspired by the activity-oriented interactive route planning system Digital Travel Mate [9], see http://www.digitaltravelmate.net, which allows for finding locations and planning routes in a fictional map by specifying kinds of holiday activity. Routes can also be interactively corrected. Our focus has been on coping with the challenges of moving away from a *hard-coded* prototype map with a small predefined set of activities to a real-world application scale decoupled from the map.

DO-ROAM consists of a graphical user interface, a data integration component and a route planning engine, see Fig. 3. The tool can be used in two complementary manners. Firstly, it can display locations where a certain activity takes place, possibly at a specified time. The locations are found using the OSM tags of the map elements, which we organised in an ontology, described in Sec. 4. We provide two alternative interfaces for the activity search. The first is a simple text-based search, similar to those existing in tools like OpenStreetMap or Google Maps. We describe the functionality of the text search in detail in Sec. 6. The second interface provides an overview of the activities

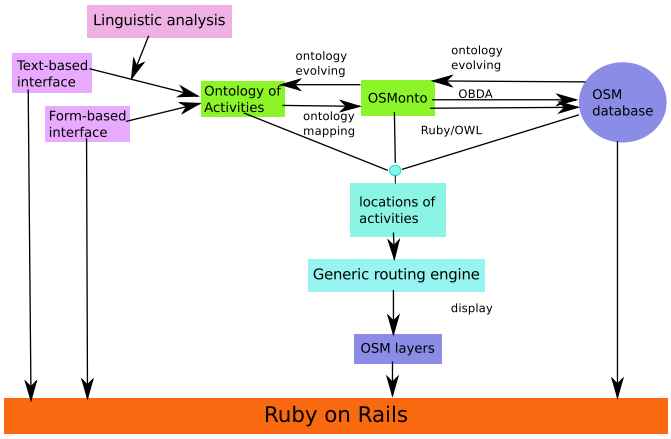


Fig. 3. Architecture of DO-ROAM

by displaying them in a tree-like structured taxonomy. Since the OSM tags representing the locations for activities are too cryptic to be presented as such, we designed an ontology of spatially-located activities to ease interaction with the user. This ontology and the way it relates to the ontology of tags are described in Sec. 5. After an activity has been specified and a restriction on opening hours selected, DO-ROAM displays markers on the map for each location where the desired activity takes place, as in Fig. 2. Moreover, the list of the names of the search results is displayed on the left side, and pop-ups providing more information on the corresponding location are available for each marker.

Secondly, the tool generates routes that include a number of locations where certain activities take place. This is done in a two stage process. At a first stage, the user must specify the starting and ending point, by either selecting a location on the map via a context menu, searching for an address or a name or directly giving the latitude and longitude. After that, the user selects a number of activities that she wants to perform along the route, in the same way as above. After each selection, the user can continue specifying activities or can proceed to the second stage to generate a route.

To find a route, we use routing engines developed by the OpenStreetMap community and available as web services (OSRM, YOURS). In the case of route planning for electric cars, we have integrated the routing engine available at http://greennav.org [2, 8], which generates energy-efficient routes (at the moment only in Bavaria). The user chooses a car type and the state of charge of the battery and the route is optimised regarding status of charge at the end of the route. Accounted for is the fact that the battery can be recharged by braking and driving downhill. The system sends the list of coordinates of the starting point, intermediate locations for activities and ending point to the routing engine, which then computes the route. The user can also specify whether the order in which the activities have been specified should be noticed or not; in the second case, the generated route will include the activities in any possible order.

As discussed in [6], there is an easy association between the tags occurring in the ontology and the database. This allows us to apply ontology-based data access (OBDA)

[7, 5] for data integration. This relies on the representation of ontologies within the Ruby on Rails framework using the library Rails-OWL also developed as part of this project, see again [6].

4 OSMonto: An Ontology of OSM Tags

OpenStreetMap's database consists of nodes, ways and relations, which can be tagged with information about the respective map element. The convention is that any user is free to introduce his own tags, but it is recommended to use existing tags and only have new ones if they are not already covered by the existing ones. The tags of the map elements are represented as (key, value) pairs. An element of the map may have multiple tags (see below for an example of an OSM node with its tags in an XML representation. This format has been developed by the OSM community. The listed tags vary from node to node).

```
<node id="834034642"
   lat="53.0871310" lon="8.8091071"
   version="7" changeset="6027662"
   user="Kerridge" uid="324245"
   timestamp="2010-10-13T09:51:39Z">
    <tag k="addr:city" v="Bremen" />
    <tag k="addr:country" v="DE" />
    <tag k="addr:housenumber" v="20" />
    <tag k="addr:postcode" v="28215" />
    <tag k="addr:street" v="Theodor-Heuss-Allee" />
    <tag k="amenity" v="charging_station" />
    <tag k="name" v="Elektrotankstelle swb" />
    <tag k="note" v="telephone reservation necessary" />
    <tag k="opening_hours" v="Mo-Fr  6:00-18:00" />
    <tag k="operator" v="swb" />
    <tag k="phone" v="+49 421 3593186" />
  </node>
```

Currently, OpenStreetMap's tags are organised and maintained through a collection of Wiki pages[4] that list the popular tags and specify their intended use. We here propose to complement this by organising the tags into an ontology, which we call OSMonto[5]. The OSMonto tag ontology is written in the OWL profile EL [3], which is a lightweight subset of OWL. The OSMonto ontology will bring the following advantages:

- an ontology provides an easier overview of the tags and their hierarchical structure than a Wiki does. Browsing the tag ontology can be done using ontology editors like Protégé;
- ontology mappings can provide different views on the tag ontology:
  - tags can be enriched with an ontological semantics by mapping existing ontologies to the tag ontology;
  - different tags that are used for the same concept (due to local differences or the evolving nature of tags) can be united to one ontological concept through a mapping;

[4] See http://wiki.openstreetmap.org/wiki/Tags and http://wiki.openstreetmap.org/wiki/Features

[5] See the project's homepage at http://osmonto.do-roam.org/ and the ontology at http://osmonto.do-roam.org/tags.owl

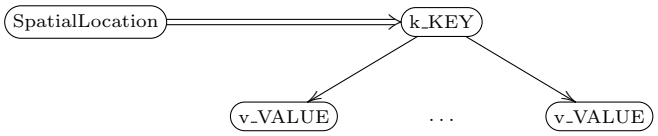
- search and navigation tools can use their own, purpose-driven ontologies (e.g. an ontology of activities that is shown to the user) and map them to the tag ontology.

The ontological perspective opens the door to *ontology-based data access* that can provide an enriched query language for the OpenStreetMap database. The ontology mappings that are necessary for obtaining the views can be generated semi-automatically or even automatically with the help of ontology matching tools. This approach provides a relatively simple, yet effective solution to the generally rather hard problem of how to relate data to ontologies.

OSMonto offers an easy and compressed overview of the keys and their values which are used in OSM. It resembles the page "Map Features" in the OSM wiki, but does not include descriptions of the tags and thus delivers a quicker overview of keys and especially their values. Also, other than the wiki page, it orientates more on the tags which are really in use at the moment (through the Taginfo website) as we only include the ontology the tags that are used with a certain frequency, see [6] for a larger discussion on this. So it is more an interesting device for users who want to make use of the existing database rather than for users interested in information how to tag something. Moreover, since all tags are in English, the ontology provides a high-level, natural-language-agnostic way of browsing the information while staying close to the original structuring of tags.

Since OSMonto is an ontology for spatial locations, we decided to introduce a class called $SpatialLocation$, around which the other classes are centred. The tags are then decomposed hierarchically according to the keys: the key becomes a superconcept of its values and we introduce an object property with domain $SpatialLocation$ having as range the class introduced for the key. Since it is possible that a key and a value have the same name whilst the names of the concepts are required to be unique in OWL (OSM has for example `station` as value of the key `railway` but also a key named `station`), we decided to prefix all keys with "k_" and all values with "v_". Notice however that some values can appear for more than one key: for example `tower` is a value both of `man_made` and `power`. In such cases we prefix the name of the value with the name of the tag (e.g. $k\_power\_v\_tower$).

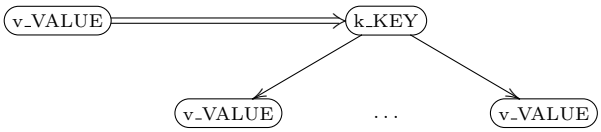
E.g., a tag `<k = "amenity" v = "charging_station">` introduces a concept $k\_amenity$ with a subconcept $v\_charging\_station$ and a role $has\_k\_amenity$ with the domain $SpatialLocation$ and range $k\_amenity$. Locations of activities are then identified using the existential restriction, in our case $\exists\, k\_amenity \bullet v\_charging\_station$. In general, the graph introduced by a tag `<k = KEY v = VALUE>` is represented below:



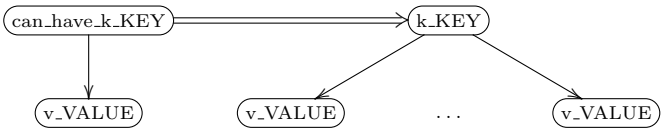
We have followed this approach whenever the value of the tag is an OSM constant rather than a string/numeral (for example, `amenity` admits as values `bank`, `cinema`, `hospital` and so on).

Another design decision is to take into account tag dependencies. For example, when a node is tagged with `<k = "amenity" v = "restaurant">` it is possible

(but not mandatory) that the cuisine of the restaurant is also tagged as <k = "cuisine" v = "seafood">. Here we distinguish two cases: some tags can be used only in the presence of a certain tag (for example, a node can be tagged with <k = "theatre:genre" v = "comedy"> only if it is tagged with <k = "amenity" v = "theatre">) but some tags can be used in the presence of more than one tag (for example, not only restaurants have cuisine, but also pubs or fast-foods). In the first case, we simply introduce an object property from the value that allows the dependency (in our case, $v\_theatre$) to the dependent tag (in our case $k\_theatre : genre$) like in the figure below:



while in the second case, we introduce a superclass of all tags that enable presence of the dependent tag (in our example $can\_have\_k\_cuisine$ is a superclass of v_bar, $v\_restaurant$ and so on) and use this superclass as domain of the object property:



Notice that it is possible that the value of a tag is not only an OSM constant (like in the case of cuisine, which can be for example chinese, italian, and also sushi or vegetarian) but a string/numeral (for example, number of rooms in a hotel). In this case, we introduce data properties, with the same two subcases as above.

A special case to consider is the tags that admit as value yes or no. For such tags we do not introduce a class with the name of the tag having a subclass v_no (or rather $k\_TAG\_v\_no$ to ensure unique names), but rather an object property with the name of the tag having as range a concept yes/no, with subconcepts v_yes and v_no. Finally, since smoking can not only be tagged as permitted or allowed, but also as allowed in dedicated, separated or isolated areas, we grouped these concepts as subclasses of a concept we called $Decision$.

In OpenStreetMap, addresses are not represented compactly, but using a number of tags (e.g. for city, country, house number, street, postal code) prefixed with addr:. The values of these tags are simply strings. There are a number of tags that enable the presence of address tags. Following the same principle as above, we grouped them as subclasses of a new concept which is then the domain of an object property $has\_address$. The range of this object property is $Address$. We then introduce for each address tag a concept $addr : TAG$ and an object property $has\_addr : TAG$ with domain $Address$ and range $addr : TAG$, and a data property relating $addr : TAG$ to the datatype $String$:

5 Enriching the Semantics of Tags

In our envisioned scenarios, activities play a central role. We aim to present them to the user as a more structured interface element for guiding their selection. Moreover, we want to allow a certain degree of flexibility by performing a lexical analysis on the free text queries of the user and trying to match synonyms of the used words with concepts of the ontology of tags. Notice that the structure of OSM tags does not always provide a clean ontological classification of related concepts, and duplications sometimes occur.

Towards these goals, we have therefore designed an ontology of activities. The concepts of the ontology refer to locations where a certain activity takes place. This provides an abstraction level from the representation of the data in the databases and thus the user can express queries using a vocabulary closer to natural language. Notice that from an ontological perspective, the ontology developed is a task ontology: here the main motivation is not to create and specify a model of a domain, but to solve a well-defined task, namely, searching locations. We refer the interested reader to see [6] for a larger discussion on the design of this ontology.

We then connect the ontology of activities to concepts in the ontology of tags via an ontology mapping.[6] This also provides a way to semantically connect related tags.

For example, when searching for a place to swim, OSM offers a wide range of tags: `<k="amenity" v="swimming_pool">`, `<k="leisure" v="swimming_pool">` as well as `<k="sport" v="swimming">`. Sometimes this occurs because of changes in the tagging system, which are not immediately taken over by the users in the data. In the ontology of activities, we can have a single concept $Swimming$ which is mapped to the derived concept $\exists\, has\_k\_amenity \bullet v\_swimming\_pool \sqcup \exists\, has\_k\_leisure \bullet v\_swimming\_pool \sqcup \exists\, has\_k\_sport \bullet v\_swimming$.

Since the number of concepts and roles is quite large, providing such a mapping manually would be a very tedious process. We can, however, use an ontology matching tool to obtain a list of pairs of concepts that are in correspondence. This approach is known to be very effective - e.g. with the ontology matcher Falcon, the degree of automation reaches 80%. However, since in our case locations are identified by existential restrictions, we do not only have to verify and confirm the matches produced by the tool, but also to add ourselves the appropriate role restriction. While we have done this manually so far, automating this process is relatively straightforward and we are in the process of developing a specialised matching tool to cope with this issue.

Example 1. In the DO-ROAM project, we created a unified concept for charging stations for electric cars. It combines the tags `fuel:electricity=yes` (fuel stations with a possibility to charge electric cars) and `amenity=charging_station` (a device solely for charging electric vehicles). The mapping then is:

$$ChargingStation \mapsto$$
$$\exists\, k\_amenity \bullet v\_charging\_station \sqcup$$
$$(\exists\, k\_amenity \bullet v\_fuel \sqcap \exists\, k\_fuel\_electricity \bullet v\_yes)$$

[6] This procedure makes it also less dependent of the specific data base (in this case of Open Street Map). A parallel connection to Google Maps e.g. is planned. This would be realised with another ontology mapping.

Example 2. The user may search not only for single activities, but classes of activities, for example, not just restaurants or fast-food, but gastronomy in general The corresponding concept in the ontology of tags is obtained by taking first the union of all subclasses of $Gastronomy$:

$$Gastronomy \mapsto Bar \sqcup Restaurant \sqcup \ldots$$

and then mapping them individually to the ontology of tags:

$$Gastronomy \mapsto \exists\, k\_amenity \bullet v\_bar \sqcup \exists\, k\_amenity \bullet v\_restaurant \sqcup \ldots$$

Example 3. We can also do more specific searches, like looking for an italian restaurant. All restaurants with an italian cuisine are italian:

$$ItalianRestaurant \mapsto Restaurant \sqcap \exists\, hasCuisineOfNationality \bullet Italian$$

and again this termed is mapped structurally to the ontology of tags:

$$ItalianRestaurant \mapsto \exists\, k\_amenity \bullet v\_restaurant \sqcap \exists\, k\_cuisine \bullet v\_italian$$

6 Query Corpus Experiment

Building an intuitive query field interface for specifying routes demands a less prescriptive and more descriptive approach to human-machine interaction. Query fields present a design challenge because they do not constrain human behaviour (guide users) as much as drop down menus, check boxes and radio boxes, thus compelling users to type what they assume the system can answer without making the available search space explicit. As we shall see in the examples, user queries are not similar to any text in books or webpages nor to any corpus of other genre. For this reason, we could not use a general purpose corpus of representative German texts and were forced to observe what users would really write most frequently in our website to create a suitable text analyser for their queries.

To collect a corpus of route queries, we conducted a preliminary controlled experiment in Bremen, Germany with 12 participants: 7 males and 5 females; all spoke German with their parents, partners and outside the family. Participants had to perform 10 tasks of planning routes with the map. For each task there was the description of a situation in which a route plan was needed (as in Example 4) and the user was required to use the query field in our website to find the desired route. The intended results were precomputed before the experiment and presented to the user on query submission independent of what the user had written in the query field. Our goal with this experiment was to verify how much variation there is in user queries when they search the same routes and what query patterns the system is expected to understand.

Example 4. Task: Du sitzt in der Bremer Kneipe “Zum Feldschlößchen” und möchtest wissen, wie Du zum Restaurant “Das kleine Lokal” in Bremen kommst.

Translation: You are at the bar “To The Little Field Castle” in Bremen and want to know how you get to the restaurant “The Little Restaurant” in Bremen.

To prevent strong priming, we avoided offering textual formulations in the situation descriptions that could be reused in a direction query by making sure that referent names such as *Zum Feldschlößchen*, *Das Kleine Lokal*, and *Peter-Weiss-Straße* were used exclusively as tokens of identifying phrases or clauses such as *in der Kneipe "Zum Feldschößchen"* (in the bar "To The Little Field Castle") and in *zum Restaurant "Das Kleine Lokal"* (to the restaurant "The Small Restaurant"), or as location relata in locating clauses such as *der in der "Peter-Weiss-Straße" wohnt* (who lives in the "Peter Weiss Street"). Referents were chosen to cover a broad set of name and entity kinds in OSM data. Name kinds includes names with or without parts such as case markers, class markers, and separators while place kinds include facilities, shops, streets, districts, and cities. A careful analysis of the route query corpus showed participants have taken one of two strategies to specify routes. The most frequent strategy consisted of writing a frame with slots for the entity names or entity kind names in the same order as they are intended to be visited. Separating the slots, we found punctuation symbols such as hyphens and colons, the English keywords "route", "from" and "to", and the equivalent German keywords "weg", "von" and "nach"/"bis". This query pattern shows most users expect websites to spot entity names or entity kind names in the query and to use the order of frame slots as the order of locations to be visited. This expectation resulted in queries such as *Route: St.-Johannis-Schule - Café - Buchladen - Hohentor* (Route: St. Johannis School - café - book store - Hohentor).

The second strategy consisted of writing a sequence of phrases in fluent German, each one carrying a case marker identifying its function in the route query (origin, destination or intermediary paths). As case markers were used, the order of query constituents was not always the same as the order of locations to be visited as one can see in *von der St.-Johannis-Schule zum Hohentor via Buchladen und Café* (from the St. Johannis School to Hohentor through bookstore and cafe).

In the task of going from The Little Field Castle to The Small Restaurant, the frame with slots was favoured by most participants as illustrated below.

Frame with Slots (5 examples out of 10)

(1) route Feldschlößchen "das kleine lokal"
(route Little Field Castle "the small restaurant")

(2) route feldschlößchen das kleine lokal
(route little field castle the small restaurant)

(3) route Zum Feldschlößchen bis Das Kleine Lokal
(route To The Little Field Castle until The Small Restaurant)

(4) Feldschlößchen to Das Kleine Lokal
(Little Field Castle to The Small Restaurant)

(5) Zum Feldschlößchen Bremen - Das kleine Lokal Bremen
(To The Little Field Castle Bremen - The Small Restaurant Bremen)

Phrase Sequence with Case Markers (2 examples out of 2)

(6) vom Feldschlößchen zum Kleinen Lokal
(from the Little Field Castle to the Small Restaurant)

(7) vom feldschößchen zum kleinen lokal
(from the little field castle to the small restaurant)

In the frame slots, on the one hand, participants have used either the nominative form of an entity name such as *Zum Feldschlößchen* and *Das Kleine Lokal* (To The Little Field Castle, The Small Restaurant) or they have used an inflexible form composed of the parts of an entity name that do not carry case markers such as *Feldschlößchen* for *Zum Feldschlößchen*. Entity names with a discontinuous case marker such as *Das Kleine Lokal* (the word *Das* and the last *e* in *Kleine*) had no inflexible form. Entity kind names such as *Café* (cafe) and *Buchladen* (bookstore) were used without articles.

When participants used sequences of phrases with case markers, on the other hand, queries were very similar to one another. Case markers were used to indicate whether a location was an origin such as *vom Feldschlößchen* for *Zum Feldschlößchen* (from The Little Field Castle), a destination such as *zum Kleinen Lokal* for *Das Kleine Lokal* (to The Small Restaurant) or a path such as *via Buchladen und Café* (through bookstore and cafe). These two strategies demand two different approaches for handling queries. For the strategy of frame with slots, – detectable by the presence of special punctuation and keywords, – names can be spotted either in their nominative or invariable form, both of which can be precomputed from the OSM database with rules. For the strategy of phrases with case markers, spotting names is not possible. For instance, the nominative form *Das Kleine Lokal* (The Small Restaurant) is not spottable in the phrase *zum Kleinen Lokal* (to the Small Restaurant). Even if spotting such names were possible, their order would not necessarily correspond to the order of locations in the route. For this reason, we intend to implement a combinatory categorial grammar to produce further processable logical forms out of such queries.

7 Conclusions and Future Work

The route planning component is further developed at the moment. Future improvements concern, for instance, better mechanisms for the reduction of the search results. At the moment, the first 35 results are shown to improve both readability for the user and the speed of DO-ROAM. Desirable would be a ranking algorithm for the results. This could be achieved by creating a user profile recording favourite search results or by using a webpage ranking like in Google Maps. We also plan to make routes modifiable in a way going beyond from route modifications in Google maps, because the place of an activity might also be changed. An interesting open research question is to remove the separation between searching locations for activities and route generation, that is, the system will find the best places for activities and the best route simultaneously.

An ontology for OpenStreetMap tags similar to OSMonto has been developed in the EU FP-7 project LOD2 [10]. This ontology does not employ our rule to leave out tags that are only rarely used; this produces a lot of noise in particular what the data properties concerns. OSMonto not only avoids this noise, but also employs roles (specified with their domains and ranges) more systematically than the LOD2 ontology does. Altogether, we think that OSMonto provides a cleaner approach, while its extraction from the OSM tags still is an algorithmic process (see the description in Sect. 4).

The potential uses of OSMonto, our ontology of OpenStreetMap tags, that we have pointed out can be realised only if the ontology is kept up to date with the current de-

velopment of OpenStreetMap tags. To ensure this, further research about manual and automatic update facilities is needed, including incorporating related work done for instance in the Web 2.0 context (see e.g. [4]). Probably an automatic update via TagInfo and the tagging wiki pages can do most of the work, keeping the number of necessary centralised manual corrections at a minimum. On the other hand, links to existing ontologies might suggest useful ontological structuring principles that need to be implemented manually. Eventually, the ontology may also give some fruitful insights into how to extend and structure the realm of OpenStreetMap tags.

As for the further improvement of the text field interface, we have invited members of the OpenStreetMap community to participate in the query corpus experiment and we shall have a larger corpus to support our automatic analysis in the near future.

Acknowledgements. We would like to thank Christian Clausen for doing implementation work and Gregor Horsinka for help with OpenStreetMap. This work has been supported by the German Research Foundation (DFG), Project I1-[OntoSpace] of the SFB/TR 8 "Spatial Cognition".

References

1. Neeharika Adabala and Kentaro Toyama. Purpose-driven navigation. In M. Andrea Rodríguez, Isabel F. Cruz, Max J. Egenhofer, and Sergei Levashkin, editors, *GeoS*, volume 3799 of *Lecture Notes in Computer Science*, pages 227–233. Springer, 2005.
2. Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. In Rüdiger Dillmann, Jürgen Beyerer, Uwe D. Hanebeck, and Tanja Schultz, editors, *KI*, volume 6359 of *Lecture Notes in Computer Science*, pages 309–316. Springer, 2010.
3. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL Envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 364–369, 2005.
4. Silvia Bindelli, Claudio Criscione, Carlo A. Curino, Mauro L. Drago, Davide Eynard, and Giorgio Orsi. Improving search and navigation by combining ontologies and social tags. In *OTM '08: Proceedings of the OTM Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems*, pages 76–85. Springer, 2008.
5. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The MASTRO system for ontology-based data access. *Semantic Web*, 2(1):43–53, 2011.
6. Mihai Codescu, Gregor Horsinka, Oliver Kutz, Till Mossakowski, and Rafaela Rau. DOROAM: Activity-oriented search and navigation with OpenStreetMap. In C. Claramunt, S. Levashkin, and M. Bertolotto, editors, *Fourth International Conference on GeoSpatial Semantics*, volume 6631 of *LNCS*, pages 88–107. Springer, 2011.
7. Antonella Poggi, Mariano Rodriguez-Muro, and Marco Ruzzi. Ontology-based database access with DIG-Mastro and the OBDA Plugin for Protégé. In Patel-Schneider, editor, *Proc. of the 4th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 DC)*, volume 496. CEUR-WS, 2008.
8. Martin Sachenbacher, Martin Leucker, Andreas Artmeier, and Julian Haselmayr. Efficient energy-optimal routing for electric vehicles. In Wolfram Burgard and Dan Roth, editors, *AAAI 2011*. AAAI Press, 2011.
9. Inessa Seifert. Region-based model of tour planning applied to interactive tour generation. In Julie A. Jacko, editor, *HCI (3)*, volume 4552 of *LNCS*, pages 499–507. Springer, 2007.
10. Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Interoperability, Usability, Applicability*, submitted. http://www.semantic-web-journal.net/content/linkedgeodata-core-web-spatial-open-data.