OSMonto - An Ontology of OpenStreetMap Tags

Mihai Codescu*, Gregor Horsinka*, Oliver Kutz**
Till Mossakowski*,**, Rafaela Rau*

* DFKI GmbH Bremen, Germany
** Research Center on Spatial Cognition,
SFB/TR 8, University of Bremen, Germany

Abstract. We provide an ontology for OpenStreetMap tags, called OSMonto, which can be used in different ways: (1) browsing of the hierarchy of tags is eased, (2) tags can be related with other ontologies, let it be tool-specific ontologies or more general ontologies like the recent Schema.org ontology supported by the dominant search engines, and (3) concepts can be unified, even if they are tagged in various ways. This will foster the integration of OpenStreetMap into the semantic web.

Keywords. OpenStreetMap tags, ontologies, Semantic Web

1. Introduction and Motivation

OpenStreetMap has evolved into a rich source of geodata. When searching and navigating through a map portal like www.openstreetmap.org, semantic metadata can greatly help with providing an intention and activity-based access to the data. In the case of OpenStreetMap, the metadata is provided in the form of tags that are entered into the database in a Social Web and wiki-like manner. Metadata obtained through such Social Web, collaborative and community based efforts have specific characteristics, namely they evolve in a bottom-up way, contain a lot of noise (typos, redundancies, etc.) and are subject to constant change.

Currently, OpenStreetMap's tags are organised and maintained through a collection of Wiki pages[1] that list the popular tags and specify their intended

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

use. In this paper, we propose to complement this by organising the tags into an *ontology*, which we call **OSMonto**[2].

This 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;
    - 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.

2. Ontologies

Ontologies are formal descriptions of the concepts in a certain domain of discourse and can be informally understood as fixing a meaning for the terms of a particular field. Ontologies are used in artificial intelligence, the semantic web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it. Domain ontologies are typically formulated in the web ontology language OWL[3].

Formally, an OWL ontology signature consists of sets of *atomic concepts*, *roles* and *individuals*, which fix the vocabulary. Sentences that can be expressed are of two types: TBox sentences are subsumption relations

[2]See the project's homepage at http://osmonto.do-roam.org/

[3]http://www.w3.org/TR/owl2-overview/

between concepts which are defined inductively from atomic concepts using the universal concept, the empty concept, unions, disjunctions, negations and universal and existential quantification over roles. ABox sentences contain assertions saying that certain individuals belong to certain complex concepts expressible in the vocabulary. Since the ontologies we use here do not contain individuals, we will concentrate on presenting TBox sentences.

Several syntaxes have been designed for ontology languages; in this paper we prefer to use Manchester OWL syntax (Horridge et al, 2006) which provides, for the fragment corresponding to the description logic ALC, the following grammar for concepts:

C ::= A | Thing | Nothing | C and C | C or C | not C | R some C| R all C

where R is a role and A is an atomic concept.

The semantics is set-theoretical: an *interpretation* I consists of a non-empty set W (the universe) and an interpretation function $.^I$ which assigns a subset of the universe to each atomic concept, a binary relation to each role and an element of the universe to each individual. The interpretation extends from atomic concepts to complex concepts in the expected set-theoretic way following the grammar, more precisely: he top concept Thing is interpreted as the universe W, Nothing as the empty set (bottom concept), a *conjunction* C and D by the intersection of the interpretations for C and D, a *disjunction* C or D by the union of the interpretations, a *complement* not C by set-theoretic complement, and finally *universal* (R all C) and *existential* (R some C) *role restrictions* as follows:

$(R \text{ all } C)^I$ = {x in W | forall y in W . $R^I(x,y)$ implies y in C^I}, and

$(R \text{ some } C)^I$ = {x in W | exists y in W . $R^I(x,y)$ and y in C^I}

Two ontologies can be related by an *ontology mapping*, sending atomic concepts, roles and individuals of the source ontology to (not necessarily atomic) concepts, roles and individuals of the target ontology. Among many other applications, ontology mappings are important for extracting modules from large ontologies.

3. OSMonto: An Ontology of OSM Tags

OpenStreetMap's internal files are lists 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 be-low 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; Sa off; Su off" />
  <tag k="operator" v="swb" />
  <tag k="phone" v="+49 421 3593186" />
 </node>
```


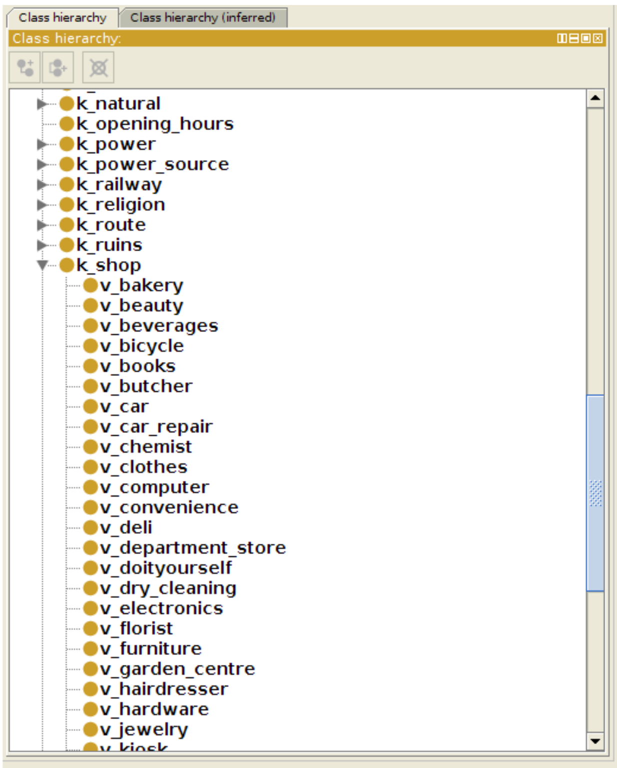

Figure 1: OSMonto: the ontology of OSM tags viewed with Protégé

The ontology of tags, OSMonto, shall stay as close as possible to the structure of the OSM files in order to facilitate ontology-based database querying. This means that we do not try to correct any possible conceptual mistakes in the taxonomy of OSM tags, but rather have it reflected faithfully in the structure of the ontology. Indeed, desired adaptions of the ontology concepts can be achieved by ontology mappings.

When designing OSMonto, it makes sense to decompose the tags into a hierarchy according to the keys: the key becomes a superconcept of its values. We have followed this approach whenever the value was an OSM constant rather then a string/numeral. 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 "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_". E.g.: k = "amenity" and v = "charging_station" would introduce a concept "k_amenity" with a subconcept "v_charging_station". Another problem is that some values are subclasses of more than one key.

E.g., "v_no" is a subclass of "k_smoking" but also of "k_smoking_outside". (We maintain our design uniform, so "v_no" must be a concept; other choices would also be available.) In this case, we extended the value to "v_smoking_k_no". Another design decision was 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 is also tagged:

k = "cuisine" v = "seafood".

In such cases, we introduce a role hasCuisine with "v_restaurant" in domain (it is also possible that "v_fast_food" is tagged with "k_cuisine") and range "k_cuisine" in order to be able to select only those restaurants with a certain cuisine.

Recalling the example of French restaurants of the previous section, notice that nation specific cuisines are added directly as subconcepts of "k_cuisine". This is conceptually a mistake in the design of OpenStreetMap's tags, which we here reflect in the ontology which is meant to be very close to the OSM tags.

In order to create a realistic ontology of OSM tags, one faces the problem of an open project where everyone is allowed to contribute - which is also an OSM strength. This has the effect that the data source can be regarded as dynamic, not only at the level of entries, but also at the level of tags. In the OSM wiki page[4], there exists a list of tags, but this list does not reflect the status quo of the actual OSM databases.

[4]http://wiki.openstreetmap.org/wiki/Map_features

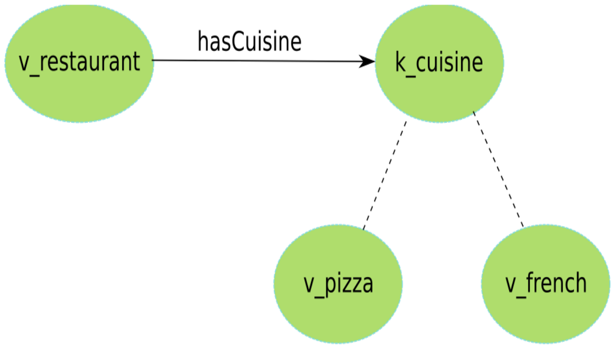

Figure 2: Restaurants with cuisine

Some tags which are in the wiki page are not yet tagged by the community, some tags which were abolished through discussion in the wiki or the mailing lists are still used by the mappers. Therefore, the wiki provides only an overview of the available tags; to have a more realistic estimation, one should use websites like Taginfo[5], where the OSM data of the whole world is searched and a list of tags in use, sorted by the number of occurrences, is provided as a result. Of course, this list will also contain spelling errors or falsely used tags.

The most straightforward solution here is to consider relevant those tags that have a certain, high occurrence in the database, using the list provided by Taginfo. This strategy could result in a limitation using a certain percentage (e.g., all values with, say, more than 0.3% occurrence rate for the respective key are included), but this approach fails to capture all interesting values in the cases where some keys appear with a far higher occurrence and thus the percentage of important values is low. Also, some keys have far more values (e.g., amenity with 7714 values in use according to Taginfo) than others (e.g., smoking with 22 values), so that the percentage of each value naturally is quite low, which is another point against a certain percentage as a limit for inclusion.

This is why a limitation based on the absolute occurrence of a value makes more sense. In our case, we decided to select all values which occur more than 100 times in the database. Spelling errors are thus excluded as well (there is never 100 times the same mistake), and still all relevant values will be in the database. Theoretically, this threshold could be exceeded by mis-

[5]http://taginfo.openstreetmap.org

takes created during automatic tagging procedures. In reality, there is no evidence in Taginfo that this is the case. It is either prevented by the professionalism of those using automatic tagging, or mistakes of such quantity are quickly noticed by the community and repaired.

After this procedure, we added the tags that are in the wiki but not covered through our search of Taginfo. This guarantees that we also include tags which are not yet used by the mappers, but in the future shall be implemented or will replace other tags. To keep this ontology of tags up-to-date, one option is to promote it within the OSM community, as we do through this paper. People creating new tags could include them themselves into the ontology as well. Another option is automation, e.g., programs searching regularly through Taginfo for new tags.

4. Applications

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). 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.

Another possible use of the ontology is the *unification of concepts*. E.g. when searching for a place to swim, OSM offers a wide range of tags. Sometimes this occurs because of changes in the tagging system, which are not immediately taken over by the users in the data. One can quite easily introduce a new concept in the ontology which contains the tags amenity=swimming_pool, leisure=swimming_pool as well as sport=swimming. In OWL notation, such a concept would be defined as follows:

```
Class: Swimming
EquivalentTo: k_amenity_v_swimming_pool or
k_leisure_v_swimming_pool or
k_sport_v_swimming
```

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).

Tags can be enriched with an ontological semantics by mapping existing ontologies to the tag ontology; interesting such ontologies would for instance be OpenCyc[6] (general knowledge base and commonsense), GUM-Space[7] (a linguistic ontology of space), or Dolce[8] (a foundational ontology for cognitive engineering). Of particular interest is the linking together of different such aspects within a network of ontologies in order to combine the different viewpoints into one coherent view (see Kutz et al. 2010).

Also, Schema.org is a recent ontology that shall be used to annotate web content, with the goal of allowing to search the Web semantically (launched jointly by Bing, Google, and Yahoo on 2nd of June 2011). Since this effort is supported by the dominant search engines, this ontology is expected to get high impact in the future. The vocabulary used in Schema.org was inspired by several earlier efforts, in particular Microformats, FOAF, GoodRelations, and OpenCyc. Via an ontology mapping, OSM can be linked to Schema.org.

In the DO-ROAM project[9], we have developed an *activity-oriented navigation tool* for OSM. This means that the user specifies a starting and an ending point for her trip, then chooses at each time an activity she wants to perform and the system provides markers on the map for the locations where the activity takes place for the user to select. Once all desired activities have been selected, the system generates a route from the starting point to the ending point which includes the selected locations for the user's activities. We have also integrated the energy-efficient route generation of the greennav.org, using external (not OSM based) data. An older prototype builds on top of the existing Rails portal for OpenStreetMap and complements the free text search of engines like Nominatim with activity search, possibly taking into account opening hours, via both the free text search and in a more guided fashion, by browsing a tree of activities.

The central element in both tools is an ontology of spatially located activities. On one side, it is presented to the user as a interface element for guiding the selection of the desired activity. Moreover, we 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. On the other side, the ontology of activities is connected with the ontology of tags via a mapping.

---

[6]See http://www.cyc.com/opencyc

[7]http://ontologydesignpatterns.org/wiki/Ontology:GUM-Space
and http://www.ontospace.uni-bremen.de/ontology/gum.html

[8]See http://www.loa-cnr.it/DOLCE.html

[9]http://do-roam.org

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 very effective - e.g. with the ontology matcher Falcon, the degree of automation reaches 80%.

This means that the user is still required to verify and confirm the matches produced with the tool, and possibly introduce new matchings between concepts that were not identified by the tool's analysis. For example, in the case of the charging stations, the matching tool is able to match the concept with the tag “charging_station”, but the semantic analysis is not powerful enough to match it with “fuel:electricity” as well and this must be provided by the user. On the other hand, since the tags for swimming places are syntactically similar, the tool would estimate all of them as possible variants with a high probability and the user can decide to create a concept for the union of the 3 variants.

Ontology-based data access is a data integration methodology that separates the `knowledge' about data from reasoning about it. This is achieved by providing an abstract representation of the application domain with the help of an ontology, a schema of the sources where the real data is stored, together with a mapping between the elements of the ontology and those of the data schema. Typically, the schema of the data is assumed to be a relational database schema, and the mapping provides a query in the database for each concept and each role of the ontology. The advantage of this approach is that we can use the knowledge base constituted by the TBox and the ABox sentences of the ontology to derive information about the data which is not present in the database, using query rewriting.

The data integration management component of DO-ROAM follows the principles of OBDA: the domain of interest - spatially located activities - is modelled as an ontology, the OpenStreetMap data is stored in a database, and the concepts of the ontology are related to queries in the database.

For the representation of and access to ontologies within the Ruby on Rails framework, we have developed a new library, Rails-OWL. Since OWL is represented in XML, our library is based on the existing library REXML[10] for reading in XML documents. Rails-OWL represents an OWL ontology in the Rails database. This allows programmers to easily and flexibly access ontologies in a way similar to the access of the geodata.

In ontology-based data access, usually, one SQL query per ontology class is designed manually, and this is used for the database interpretation of ontology terms, implemented by query rewriting. In case of OpenStreetMap, we

[10] http://www.germane-software.com/software/rexml/

would need to design dozens of such SQL queries, which is a tedious process. Instead, we use the OSM tag ontology, which is tailored towards the OSM database in such a way that the relation between classes in the OSM tag ontology and the OSM database is generic: since the basic classes directly correspond to keys and values of OSM tags, the corresponding SQL queries are simple, and this is then used for query rewriting of more complex class terms. This query rewriting is implemented in Rails-OWL easily, because classes, roles and such are first-class citizens.

5. Conclusions

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 development 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. Bindelli et al. 2008). Probably an automatic update via TagInfo and the tagging wiki pages can do most of the work, keeping the number of necessary 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.

References

Horridge M, Drummond N, Goodwin J, Rector A, Stevens R, and Wang H (2006) The Manchester OWL Syntax. OWL: Experiences and Directions (OWLED-06), CEUR-WS Volume 216

Kutz O, Normann I, Mossakowski T, and Walther D (2010) Chinese Whispers and Connected Alignments. Proc. of the 5th International Workshop on Ontology Matching, OM-2010 (ISWC-2010)

Bindelli S, Criscione C, Curino C A, Drago M L, Eynard D, and Orsi G (2008) Improving Search and Navigation by Combining Ontologies and Social Tags. On the Move to Meaningful Internet Systems (OTM-08), p. 76-86, LNCS 5333, Springer

Codescu M, Horsinka G, Kutz O, Mossakowski T, and Rau R (2011) DO-ROAM: Activity-Oriented Search and Navigation with OpenStreetMap. Christophe Claramunt, Michela Bertolotto, Sergei Levashkin (Eds.), Fourth International Conference on GeoSpatial Semantics (GeoS-11), p. 88-107, LNCS 6631, Springer