Finding Apartments with Linked Open Data

Raimund Schnürer

Institute for Geoinformatics (ifgi)
Weseler Str. 251, 48151 Münster, Germany

Abstract. This paper outlines how searching apartments in the Internet can be improved by Linked Open Data. Exemplary, an apartment is described with existing vocabularies and a proprietary schema. Sample queries over the assembled data illustrate potential use cases. To automatically generate further triples, some inference rules are suggested. Challenges, modeling alternatives and possible extensions of the dataset are briefly identified in the last section.

Keywords: Apartments, Linked Open Data, RDF, Search, SPARQL

This is an exercise report and not an official research paper.

1 Motivation

Imagine you have successfully applied for an internship in a foreign country and you are searching now an apartment there. For this, you would probably use one of the many apartment finders in the Internet, for example Immobilienscout24[1] in Germany. During your search though, it is likely to encounter several difficulties. Since most of the apartment finders are specialized to a certain country, they are kept only in local language. Non-native speakers, who are not familiar with this language, will have severe problems translating the offers. Even more challenging to understand are local terms. Kaltmiete[2] is an example of such a term in German.

Linked Open Data helps to overcome those hurdles. When labels of resources are given in different languages, it is easy to switch between them. Furthermore, users can simply find out unknown semantics by following the links and have a look on the description and comments there. One other major advantage of Linked Open Data is that different datasets can seamlessly be interrelated. In terms of apartments, you could for example inform yourself about crime statistics of the area the apartment is in – assuming this data is available for the public.

But not only apartment seekers, also apartment providers could benefit from using Linked Open Data. In contrast to conventional apartment finding websites, which are restricted to a fixed set of attributes, Linked Data allows a flexible vocabulary. For example, my dataset contains a triple for the flooring of an apartment. Alternatively, free-text forms are offered on traditional websites. However, searching capabilities

---

[1] http://www.immobilienscout24.de/

[2] Kaltmiete (lit. cold rent) is the part of the rent which covers only the room use [1]

are here limited to a word-by-word search. In contrast to this, Linked Open Data can be queried about with SPARQL. SPARQL is a querying language similar to SQL. One exemplary query can be found in chapter 3 of this paper.

2 Description of my dataset

My dataset can be accessed online at http://ietherpad.com/apartment. All of the triples were manually collected. They are currently stored in N3-Notation, but with any23[3], they can be easily converted into any another Linked Data format, such as Turtle or RDF/XML.
A special focus has been laid to reuse existing vocabularies. Altogether 16 different vocabularies were used. Their namespaces are right at the beginning of my triple store. Their specifications are listed at the end of this paper. The application of each vocabulary is illustrated throughout this chapter.
Only when there did not exist any convenient vocabulary, own classes and properties were defined. Thus, my dataset consists of Data Model Definition oriented Linked Data as well as Data Model Instance Data. [2] As an instance, an imaginary apartment located in the 6th floor of the Institute of Geoinformatics, Münster was chosen.
For my topic, non-information resources are apartments in the real world and information resources are triples on my Etherpad describing those apartments. Ideally when entering the URI in the first case, one should be redirected to latter one. [3] For a better machine-interpretation, data types (e.g. ^^xsd:date) were attached to literals. For textual descriptions, language tags were used (e.g. @de, @en).

2.1 Metadata

The first triples in my dataset are about metadata. With those, the user gets a quick overview about the dataset, without going in further detail. For instance, title, primary topic and a little abstract can be found in this section. I oriented therefor on the Meta-vocab[4] from the VoCamp Wiki. There, it is recommended to use vocabulary from *Dublin Core* for this purpose. This goes along with the ISO 19115 metadata standard. So, information about the author, supported languages, creation and modification dates, target audience and target area can be expressed. The licensing aspect is covered by the *Creative Commons* vocabulary. This is required to clarify the legal situation how the data can be used.

2.2 Spatial Properties

The next triples in my dataset are related to space. To quickly locate an apartment, WGS84 coordinates are taken from W3C's *Basic Geo Vocabulary*. For a detailed representation of the floor plan of an apartment, the geometry is coded as GML

---

[3] http://any23.org/
[4] http://vocamp.org/wiki/Meta-vocab

String. A similar approach can be found on *Ordnance Survey Linked Data* and on *GeoLinked Data (.es)*.
It would not have been feasible to model single points or lines as own entities, as it is the case on *GeoLinked Data (.es)*. For an apartment seeker, this would be more confusing than helpful. Also, spatial relationships to other apartments in the same building do not matter in many cases. Moreover, it was waived to model the shapes of single rooms of an apartment. This would go more in an architectural direction rather than a geographic one. Instead, more common spatial attributes were focused.
To those certainly belongs the address of an apartment which can be described with Google's *Data Vocabulary*. Alternatively, W3C's *Contact Vocabulary* is applicable. The latter vocabulary is used later for addresses of organizations offering apartments. Size and height of an apartment can be specified with the help of the *GoodRelations* language. Unlike in chapter 2.6 of [4], anonymous entities were avoided for value and unit of measure. Address as well as size and height are announced with corresponding properties in the *Open Time and Space Core Vocabulary Specification*. Points of interests (e.g. banking houses or supermarkets) nearby the apartment are described with the near relation of this vocabulary. The floor in which an apartment lies is stated in my own vocabulary.

2.3 Temporal Properties

In this section, properties of apartments which are related to time are presented. Persons in search of apartments surely want to know when the apartment is available for renting and how long the actual offer is valid. Again, *GoodRelations* provides convenient terms for this scope. For more advanced temporal relations, I decided to take the *Time Ontology in OWL*. Some tenants might prefer apartments at particular sunshine times, for example only in the evening hours. That is why I included a property for direct sunlight exposure times. Even more important are the periods when it is possible to visit an apartment. At those times, interested people can have a look at the offered apartment and the owner can gain a first impression of potential tenants. Lastly, due to lack of existing vocabularies, I expressed the last refurbishment of an apartment in a self-made relation.

2.4 Further Properties

This subchapter covers all other attributes for describing an apartment. These are as important as the beforehand mentioned space and time properties. With *FOAF Vocabulary*, it is for example possible to easily attach links to images of an apartment. With the *GoodRelations* language, the quality of an apartment can be characterized. Having extended *GoodRelations* by own classes, costs like the monthly rent, purchase price or a bail can now be specified. Furthermore, it can be expressed which services are included in the rent (e.g. floor lightening) and which are not (e.g. warm water).
Following in the dataset, contact details for the exemplary apartment are given. A contact can either be a person or an organization. For a person, *FOAF* relations were taken - such as the first name, last name, phone and mailbox. The organization was

described with W3C's *Contact Vocabulary*. Here, further attributes like fax, homepage and address are specified. In combination with *GoodRelations* vocabulary, the opening hours of the apartment offering organization can be stated.
After that, I built triples according to my proprietary vocabulary. Diverse apartment finding websites inspired me in adding those relations. Their names should be self-explanatory: Properties having a Boolean range are smokingAllowed, petsAllowed, hasDSLConnection, hasTVConnection, isFurnished and handicappedAccessible. Properties, where objects can be arbitrary resources, are flooring (e.g. laminate), heating (e.g. oil heating), attachment (e.g. a terrace), apartmentType (e.g. penthouse) and equipment (e.g. curtains). For the properties attachment and equipment multiple usages are possible.

2.5 Rooms and Buildings

Next, a way had to be found to state which rooms are in an apartment and how many there are of them (e.g. two bathrooms, one kitchen, three bedrooms). This is a common indicator of many apartment finders in the Internet. Additionally, a property for equipment in particular rooms had to be defined (e.g. washing machine in the bathroom, microwave in the kitchen, wall system in the bedroom). Both things were realized in my own vocabulary. To indicate that a room is part of an apartment, I created a subclass of the insideOf relation from the *Open Time and Space Core Vocabulary*.
The same applies for the building the apartment is in. Potential tenants might not be only interested in the apartment, but also in the building. For example, elderly people might want to know if the building has an elevator. Also, the total number of floors in a building might be relevant. This number can be related to the floor the offered apartment lies. A significant temporal relation of a building is the year of the last refurbishment. This property could give a hint about the state of water pipes, heating and isolation. Another temporal attribute is the year of construction. From this, the style of the building can possibly be deduced. Beyond that, some images of the building can be included.

2.6 Own Classes and Properties

The last part of my dataset contains Data Model Definition oriented Linked Data. As vocabulary, I mainly used *RDF Schema*. This enabled me to define own classes (e.g. Apartment) and properties (e.g. floorNumber) which I could not find in other vocabularies. In some cases, I declared subclasses, for example PurchasePrice from *GoodRelations'* class PriceSpecification. This allowed keeping properties from the superclass.
For all properties, I defined domain and range. In the majority of cases, the domain is the Apartment class. For the range, I chose either to XML Schema datatypes (e.g. boolean, positiveInteger or gDate) or arbitrary RDF resources. All established classes and properties, I gave a label, a short comment and indicated the source. If appropriate, I used the *OWL* sameAs relation to reference similar resources.

3 Querying

My apartment dataset allows different possibilities for querying. On the one hand, you can query *within* the dataset about temporal, spatial and other attributes (or combinations of them). These three examples shall illustrate this:

- Select all apartments which are available in Münster from next month on.
- Search for apartments with at least two bathrooms, a size of more than 100m² and where pets are allowed.
- List all apartments which are higher than the fifth floor and whose building has an elevator.

The second example corresponds to the following SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX tisc: <http://observedchange.com/tisc/ns#>
PREFIX gr: <http://purl.org/goodrelations/v1#>
PREFIX ap: <http://ietherpad.com/apartment#>

SELECT ?apartment
WHERE {
   ?apartment <rdf:type> <ap:Apartment> .
   ?room <ap:insideOfApartment> ?apartment .
   ?room <ap:roomType> <dbpedia:Bathroom> .
   ?room <ap:roomCount> ?roomCount .
   ?apartment <tisc:areasize> ?size .
   ?size <gr:hasUnitOfMeasurement> "MTK" .
   ?size <gr:hasValue> ?sizeValue .
   ?apartment <ap:petsAllowed> ?petsAllowed .
   FILTER (?roomCount >= 2 && ?sizeValue > 100 && ?petsAllowed)
}
```

This request was run over the dataset with Twinkle[5], a SPARQL query tool, and returned “TestApartment1” as a result.

On the other hand, you can combine my apartment triple store with others and query *across* these datasets. Three exemplary queries are given here:

- Detect all apartments that may be affected by natural disasters (e.g. earthquakes, floods, fires).
- Identify apartments which are close to interesting places (like bars, sights, schools, hospitals, stations of public transportation, parks).
- Find apartments which are far from noisy places (such as football stadiums, rock concerts, student houses) or from a certain plant species (due to an allergy).

[5] http://www.ldodds.com/projects/twinkle/

4 Reasoning

To automatically infer triples, some reasoning can be applied to the dataset. In the following, this will be demonstrated by three cases. The first one of those takes the transitivity of subclasses into account:

IF A contains the triple ?a <rdf:type> <ap:Apartment>
THEN A also contains ?a <rdf:type> <gr:Offering> and ?a <rdf:type> <tisc:Object>

This means that all instances of the Apartment class are automatically instances of the class Offering from *GoodRelations* and the class Object from the *Open Time and Space Core Vocabulary*. ‘A’ represents all triples in my dataset.
The next two cases concern temporal and spatial attributes and involve some simple arithmetic operations. Since aggregate functions are not specified in SPARQL yet [5], these are expressed only informally: For each apartment, a triple could be generated which is the sum of all direct sunshine exposure times of this apartment during a day (e.g. 4 hours and 2 hours make 6 hours altogether).
For all near properties between an apartment and a point of interest, the Euclidean distance could be calculated (e.g. that the nearby supermarket is 1 km away from the apartment). This assumes that geographic coordinates are known for both locations. Then (if not using reification), four distance triples with starting point, ending point, value and unit could be created for each near predicate.

5 Challenges and Future work

Since not many people are familiar with Linked Open Data yet, a user-friendly interface is needed for offering and querying apartments in Linked Data format. For example, this would help a lot when creating triples according to the *Time Ontology in OWL*.
In addition to representing the geometry of an apartment as GML String, the proposed *NeoGeo Vocabulary*[6] could be taken. With this, a more detailed model – with for instance interiors and exteriors - would be possible. Adding a “below” and “above” relation to the *Open Time and Space Core Vocabulary* could facilitate modeling apartments on several floors.
In general, there is the discussion of introducing an upper level ontology to Linked Open Data [6]. This kind of ontology could provide a systematic and formal description of the Linked Open Data Cloud and describe the knowledge base at a very abstract level. It would expedite querying capabilities and inconsistencies could be better detected. I would appreciate this approach since my dataset consists of many different vocabularies which I imagine will be difficult to query about.
The apartment vocabulary itself could be further extended by including houses, student houses, flat shares, office space, and holiday apartments.

[6] http://geovocab.org/doc/neogeo.html

Used Vocabularies

Basic Geo Vocabulary http://www.w3.org/2003/01/geo/
Contact Vocabulary http://www.w3.org/2000/10/swap/pim/contact
Creative Commons http://creativecommons.org/ns
Data Vocabulary http://www.data-vocabulary.org/
DBpedia http://dbpedia.org/About
Dublin Core Metadata Initiative http://dublincore.org/documents/dcmi-terms/
FOAF Vocabulary http://xmlns.com/foaf/spec/
GeoLinked Data (.es) http://geo.linkeddata.es
GoodRelations http://www.heppnetz.de/ontologies/goodrelations/v1
Open Time and Space Core Vocabulary http://observedchange.com/tisc/ns/
Ordnance Survey Linked Data http://data.ordnancesurvey.co.uk/
OWL Web Ontology Language http://www.w3.org/TR/owl-ref/
RDF and RDF Schema http://www.w3.org/TR/rdf-schema/
Time Ontology in OWL http://www.w3.org/TR/owl-time/
XML Schema http://www.w3.org/2001/XMLSchema

References

1. Wikipedia: Kaltmiete (2011). Available at: http://de.wikipedia.org/wiki/Kaltmiete
2. Idehen, K.U.: State of the Linked Data Web (2010). Available at: http://www.openlinksw.com/dataspace/kidehen@openlinksw.com/weblog/kidehen@openlinksw.com%27s%20BLOG%20%5B127%5D/1455
3. Bizer, C., Cyganiak, R., Heath, T.: How to Publish Linked Data on the Web (2008). Available at: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/
4. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. In: Journal of Web Semantics 6 (3), pp. 203-217, Elsevier (2008)
5. Kjernsmo, K., Passant, A.: SPARQL New Features and Rationale (2009). Available at: http://www.w3.org/TR/sparql-features/#Aggregates
6. Jain, P., Hitzler, P., Yeh, P.Z., Verma, K. Sheth, A.P.: Linked Data is Merely More Data. In: Linked Data Meets Artificial Intelligence. pp. 82--86, AAAI Press (2009)