FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Transferring Open Government Data into the global Linked Open Data Cloud

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Christian Weiss
Matrikelnummer 0325829

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: o.Univ.Prof. Dipl.Ing. Dr. A Min Tjoa
Mitwirkung: Mag. Dipl.-Ing. Dr. Amin Anjomshoaa

Wien, 22.07.2013

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Transferring Open Government Data into the global Linked Open Data Cloud

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Christian Weiss
Registration Number 0325829

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: o.Univ.Prof. Dipl.Ing. Dr. A Min Tjoa
Assistance: Mag. Dipl.-Ing. Dr. Amin Anjomshoaa

Vienna, 22.07.2013

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Christian Weiss
Puchsbaumgasse 15/16, 1100 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Abstract

The Semantic Web aims to convert the World Wide Web into a ”web of data” by adding machine-readable semantic annotations. It is one of the most important movements regarding the World Wide Web. The concept was introduced back in 1998; however the initial ideas and plans proved to be too optimistic and the Semantic Web is not realized on a broader basis as of today.

Linked Open Data (LOD) stands for data published on the web in an open and machine-readable format that is interlinked with other open data sources. Availability of LOD is a major requirement for the realization of a global Semantic Web vision. Initiatives like DBpedia and the Linking Open Data community project catalyzed the expansion of Linked Open Data on the web. Principles were established to ensure a certain level of data quality and usefulness of published data. A wide variety of data sources is already included in the global Linked Open Data cloud.

National and regional governments around the world also began to make data openly available on the web. For example, the city government of Vienna (Austria) has started to publish datasets as part of their Open Government Data (OGD) initiative. These datasets were published in numerous formats but they are isolated and not yet designed to conform to Linked Open Data principles. The Vienna OGD initiative intends to release LOD at some point but will coordinate this effort with other OGD initiatives in Austria. Therefore the release of LOD is currently in a very early planning stage.

The goal of this thesis is to show how a subset from the Vienna OGD initiative can be converted into usable LOD. By introducing a common schema and interlinking the isolated datasets new use cases like complex querying will be made possible. The resulting data has to conform to established LOD principles and will be published on a web server so it can be accessed and queried using SPARQL.

Kurzfassung

Das Semantic Web hat das Ziel, das heutige World Wide Web durch hinzufügen von Metadaten maschinenlesbar zu machen. Es ist einer der wichtigsten Entwicklungsschritte in der Geschichte des World Wide Web. Das Konzept existiert bereits seit 1998, die ursprünglichen Ideen stellten sich allerdings als zu optimistisch heraus und die Vision ist bis heute nicht auf breiter Basis realisiert.

Linked Open Data (LOD) bezeichnet Daten, die unter Verwendung eines offenen und maschinenlesbaren Formates im Web publiziert werden und mit anderen offenen Datenquellen verlinkt sind. Die Verfügbarkeit von LOD ist eine der wichtigsten Voraussetzungen für die Realisierung des Semantic Web. Initiativen wie DBpedia und das Linking Open Data community project haben die Verbreitung von Linked Open Data deutlich vorangetrieben. Richtlinien für Datenqualität und Inhalt von offenen Daten sind mittlerweile verfügbar und eine Vielzahl von verschiedene Datenquellen ist bereits in der Linked Open Data Cloud enthalten.

Regierungen in verschiedenen Regionen der Welt haben ebenfalls begonnen, Daten offen über das Internet zu publizieren. Die Stadt Wien z.B. hat Datensätze im Rahmen ihrer Open Government Data Initiative (OGD) veröffentlicht. Diese Datensätze wurden in verschiedenen Formaten online gestellt, sind allerdings untereinander isoliert und wurden nicht nach den Richtlinien für Linked Open Data entworfen. Seitens der Wiener OGD Initiative ist die Veröffentlichung der Daten in LOD geplant, durch die angestrebte Koordination mit anderen OGD Initiativen befindet sich dieses Vorhaben aber noch in einer frühen Planungsphase.

Das Ziel dieser Diplomarbeit ist es, anhand einer Auswahl von Daten der OGD Initiative der Stadt Wien zu zeigen, dass diese in Linked Open Data konvertiert werden können. Die konvertierten Daten müssen dabei den Richtlinien bzgl. LOD entsprechen und werden auf einem Webserver veröffentlicht um neue Anwendungsfälle wie komplexe Abfragen mittels SPARQL zu ermöglichen.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Information represented on the World Wide Web is usually designed to be interpreted by humans. Recent developments were mostly focused on either content generation (e.g. Web 2.0) or content representation (e.g. HTML5). HTML5 introduces several elements to outline some limited semantics of a web page [55] but none of these developments comprehensively target a very promising use case: The interpretation of content by machines.

Since information on the web is usually not semantically annotated, it cannot be interpreted by machines. Therefore, search engines are only able to find results based on text matching but they don't really "understand" what the result is about. The Semantic Web introduces the kind of semantic annotation that is required to enable automated reasoning based on data on the web.

The Semantic Web Vision was introduced in 1998 when Sir Tim Berners-Lee published a roadmap [27] that summarizes the concept and idea. The idea is to enable machines to semantically understand content on the World Wide Web. This is realized by adding metadata to markup content. Furthermore, metadata is based on schemata, so called ontologies, enabling computational reasoning based on data that was published on the web.

In a first step this leads to a better categorization of web content. For example, when a website is categorized as being about "Pizza" and an applicable ontology states that "Pizza" is "Food", it can now be automatically reasoned that the website is food-related, even though this is not stated explicitly within the websites metadata.

The Semantic Web relies on three main concepts:

Metadata Metadata annotates and describes the data published on the World Wide Web. After adding metadata, content becomes machine-readable.

Ontologies Metadata is based on ontologies that describe hierarchies and connections between elements. A vocabulary is defined that builds the basis for the Semantic Web.

Reasoning Based on the metadata and ontologies, reasoning can be applied to generate new data.

However, the Semantic Web also relies on another very important issue: Availability of open data. High-quality interlinked data is required as a foundation to make Semantic Web applications possible.

The introduction of Linked Open Data (LOD) tries to improve the lack of data that Semantic Web applications can build upon.

Principles of linked data were first outlined by Berners-Lee in 2006 [19]. These principles are based on

- Using HTTP URIs to identify things and make them referable
- Using open standards defined by the World Wide Web Consortium (W3C) to describe things
- Including links to other resources

LOD principles are described in chapter 3.

The LOD cloud, a graphic representation of linked data on the web [10] already contains data from several sources. This cloud is already very extensive and is shown in Figure 1.1.

There are cases where data added to the Linked Open Data cloud was not originally published in a semantic data format but converted from some source. A well-known example for this approach is DBpedia, a community project that aims to extract structured information from Wikipedia and makes this information available as linked data on the web [2].

The city government of Vienna has also published datasets as part of their Open Government Data (OGD) initiative [31]. These datasets were published in numerous formats but they are isolated and were not designed not conform to established LOD principles.

The goal of this thesis is to convert a subset from the Vienna OGD initiative into LOD, conforming to the mentioned LOD principles. This includes using well established vocabulary as well as enhancing the data quality and adding links to other LOD sources.

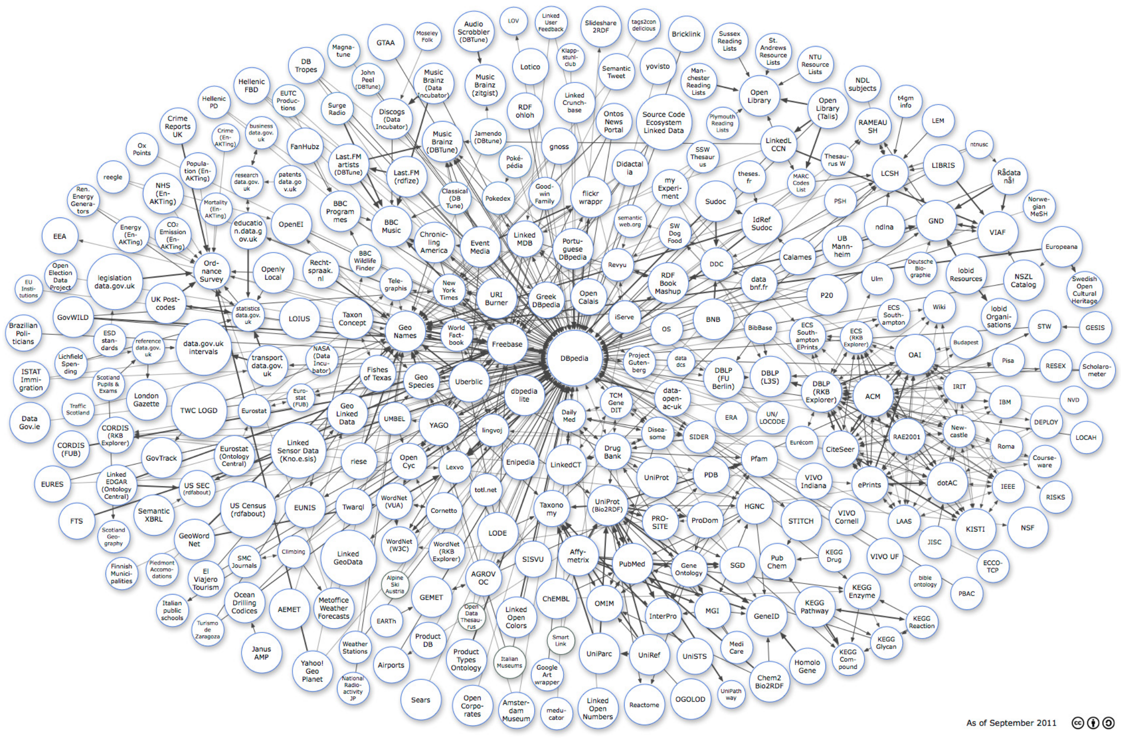


Figure 1.1: A graphical representation of Linked Open Data on the web (Source: [10])

Data formats and technologies for the Semantic Web that will be used during the thesis work are already standardized and well established (e.g. XML/RDF/OWL/SPARQL) and several frameworks exist to support the handling of large Semantic Web datasets. Standards regarding the annotation of Linked Open Data are also emerging, for example the Dublin Core schema [18]. In addition to data formats, data quality requirements for LOD have to be taken into account.

1.2 Structure of the Thesis

In this section the reader gets an overview about the motivation and how the thesis is structured. Chapter 2 introduces the key concepts, technologies and basic principles like RDF, OWL and SPARQL that build the foundation of the Semantic Web, Linked Open Data and the thesis work.

Chapter 3 is devoted to the state of the art in Linked Open Data. There will be a review of established principles that have to be taken into account when creating LOD as well as the description of examples and an evaluation regarding Open Government Data. DBpedia is a very popular source of Linked Open Data that will also be used to link the government data from Vienna to a LOD source and will there be evaluated in more detail. Other OGD initiatives will also we discussed.

Chapter 4 describes the suggested solution and the conversion of Vienna OGD into Linked Open Data. Requirements for the resulting dataset will be defined and there will be a discussion of how the conversion process is implemented and how the guidelines regarding LOD are followed. There will be an in-depth description of the data structure that was designed for the transformation as well as a discussion of technologies and frameworks that are utilized.

Chapter 5 evaluates the resulting Linked Open Data on the basis of the requirements defined before. Examples for use cases that could be developed based on the LOD dataset will be presented and the RDF graph will be reviewed and visualized.

Chapters 6 and 7 provide an outlook to future work as well as critical reflection on the thesis.

The ontology that was created for the transformation is listed in an Appendix to this document.

CHAPTER 2

Basic Principles

2.1 The Semantic Web

Information represented on the web is usually designed to be interpreted by humans, making the web highly dependent on human interaction. For example, someone who wants to book a trip to some warm country during Christmas holidays usually has to visit several websites, interpret, retrieve and combine information. It is currently not possible to instruct some computational engine to ”book a trip to some warm country during Christmas holidays”. Even when narrowed down to search tasks it is currently not possible to search for ”vacation package to a warm country during Christmas holidays”.

Machines are currently not able to combine information found on distinct websites. For example, there may be a website that lists a vacation package for South Africa and another website that states that South Africa can be considered a ”warm country”during Christmas holidays. It is easy for humans to combine this information but machines cannot interpret this data semantically and therefore fail to deliver a result.

In an article published in Scientific American in 2001, Tim Berners-Lee and his co-authors wrote [3]:

> The Semantic Web will bring structure to the meaningful content of web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users. [...] The Semantic Web is not a separate web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

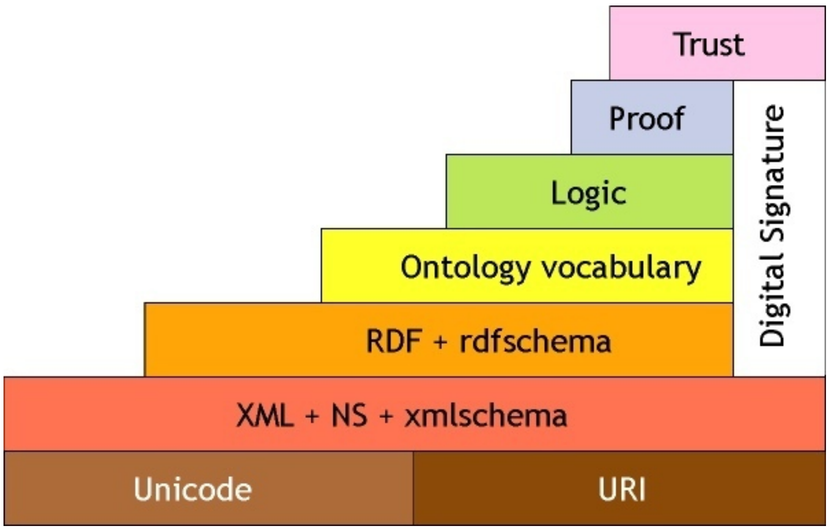


Figure 2.1: The Semantic Web layer cake (Source: [56])

The idea is to achieve this meaningful content by publishing machine-readable metadata (that may be invisible to humans) in addition to human-readable data on the web. The existing information on the web therefore becomes annotated and can be interpreted by machines.

2.2 Semantic Web Technologies

The Semantic Web consists of several technologies that can be divided into 7 layers. The so-called layer cake, as seen in figure 2.1 was introduced by Tim Berners-Lee back in the year 2000 [56].

Unicode / URI The first layer handles several low-level requirements. Unicode is a global encoding standard that is based on ASCII but contains characters for several languages as well as mathematical formulas. Unicode is used for encoding in all layers above. URI stands for Universal Resource Identifier. It is a concept that assigns a unique identifier to each and every resource used. Everything within the Semantic Web can be identified using a URI. For example, the URI for a certain company could be the URL of its homepage. URIs are used in RDF-graphs and ontologies.

XML / XML-Schema XML (Extensible Markup Language) is used to represent structured data. The markup language only provides an alphabet but no vocabulary or semantics whatsoever. XML builds the foundation of the current web in the form of (X)HTML.

Since Unicode, URI and XML are very basic technologies related to a much broader context than just Semantic Web or Linked Open Data they are not described in more detail. The more specific technologies utilized for the Semantic Web are covered in the third layer and above:

RDF / RDF-Schema RDF is the primary data format used for the Semantic Web. RDF is a triple-based format where every statement (triple) consists of subject, predicate and object. Serialization in XML is possible, however there are also other ways to represent RDF data. The object of any triple can be defined directly (e.g. being a literal) or refer to another resource (URI) on the web. RDF-Schema (RDFS) allows to create simple schemata (ontologies) for RDF.

Ontology Ontologies are used to classify information that is published in RDF. It is possible to create structures, vocabularies and hierarchies in ontologies. This is helpful to introduce semantics into RDF data. Since RDFS is not expressive enough for many use cases, the more complex Web Ontology Language (OWL) can be used instead.

Logic Based on RDF data and the semantics introduced with ontologies, it is possible to create new information out of existing data. This is called reasoning and happens within the logic layer of Semantic Web. There are specific algorithms and software tools (reasoners) to support this step. Agents that answer user requests can be considered reasoners and also work on this layer.

Proof The proof layer is used to trace the reasoning steps and verify the results.

Trust The trust layer was introduced to verify data and identify its source. Technologies and applications working on layer are largely unrealized as of today.

The lower layers (RDF, ontology and logic) are already well standardized and used for several applications. The upper about proof and trust are currently topics of research but it seems that there are not many real-world applications available.

2.3 RDF

Using RDF to represent information

RDF was selected and designed as the data format for the Semantic Web by the World Wide Web Consortium (W3C). The W3C published a first specification of the RDF data model and the XML serialization syntax as a recommendation in 1999. [41]. It was therefore already included in the earliest articles and papers describing the Semantic Web Vision.

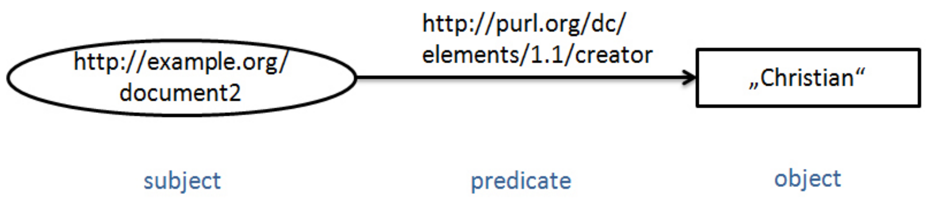

http://example.org/document2
http://purl.org/dc/elements/1.1/creator
„Christian“
subject
predicate
object

Figure 2.2: A simple RDF graph

According to the W3C, the design goals for RDF were [7]:

1. having a simple data model
2. having formal semantics and provable inference
3. using an extensible URI-based vocabulary
4. using an XML-based syntax
5. supporting use of XML schema datatypes
6. allowing anyone to make statements about any resource

Since RDF is a W3C recommendation, many Semantic Web applications use it as basis.

RDF is built upon so-called triples. A triple consists of a subject, predicate and an object. In the case of RDF, the subject names the resource described by the triple. The predicate names what the description is about (the property) and the object holds its value. A triple is also often called a statement. It is important that subject as well as predicate should be URIs that can be uniquely identified. Using this structure, almost anything can be described.

Triples can be easily visualized as a directed graph, showing some similarities to entity-relationship models that are used to describe databases. The resulting graphs are called RDF graphs. A resource is usually visualized as a circle, a value/literal as a rectangle and a predicate is visualized as a directed arc.

The example in figure 2.2 shows a simple RDF triple that describes the creator of a document. More specific, it is stated that the creator (using the standardized predicate http://purl.org/dc/elements/1.1/creator from the Dublin Core [18] metadata element set) of a certain document (URI: http://example.org/document2) is called ”Christian”.

This statement can also be expressed in serialized XML notation:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
        <rdf:Description rdf:about="http://example.org/document2">
                <dc:creator>Christian</dc:creator>
        </rdf:Description>
</rdf:RDF>
```

When RDF was specified, XML was defined as the format to serialize it. However, other options have emerged. One of them is the N3 notation, where our example would look as follows:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://example.org/document2> dc:creator "Christian" .
```

N3 notation is in general just a more compact form to display an RDF graph and is also easier to read and write while the XML notation is more compatible due to the well-known XML format. Therefore the XML serialization is especially important for exchange between applications.

Other common formats of serializing RDF that are not described in detail here are N-Triple, JSON and Turtle.

There is another notation element in RDF that is not yet mentioned: The blank node. A blank node is node without a URI that can be used for unnamed structured objects. Blank nodes are also called anonymous resources. Despite not having a URI, such a node can still be the subject of other triples. This is shown in figure 2.3, where a blank node is used to aggregate metadata, again using predicates from the Dublin Core metadata element set.

When representing values such as integers, floating point numbers and dates, datatypes are used by RDF. [7].

To summarize, a subject can either be a URI or a blank node, a predicate is always a URI and the object can be a URI, a blank node or a string literal encoded in Unicode.

Advantages of RDF

Many websites are based on data that is stored in relational databases. In fact almost all Content Management Systems (CMS) on the market use a conventional relational SQL database in the

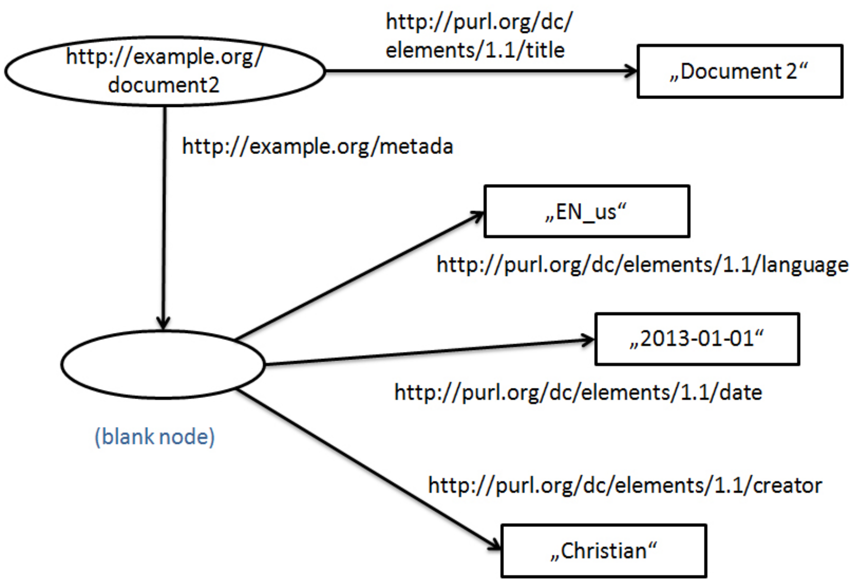

http://example.org/
document2
http://purl.org/dc/
elements/1.1/title
„Document 2“
http://example.org/metada
„EN_us“
http://purl.org/dc/elements/1.1/language
„2013-01-01“
http://purl.org/dc/elements/1.1/date
(blank node)
http://purl.org/dc/elements/1.1/creator
„Christian“

Figure 2.3: A blank node is used to aggregate some metadata

background. It is of course possible to store the data from the examples above in a relational database without using RDF. The advantage of RDF is not obvious for these scenarios at first. Semantic Web applications may even use relational databases in the background to persistently store RDF data internally. Anyway, using RDF offers several advantages over using a conventional database with a relational data schema.

By using the RDF data format, it is very easy to combine triples with statements from other sources (based on the same data model). It is also easy to add data that was not considered when the model was created. For example, adding the creation date to our document example can be done without changing the data model. More importantly: it is possible to reference resources that are stored anywhere else on the web by just using their URIs.

This also means that there is no limitation to use only data from a services own data sources. Data from other sources can easily be integrated; in fact the whole web becomes a single connected data source. Connecting relational databases is a very complicated process that cannot be applied to a global web of data.

Another advantage is that RDF is built upon basic web technologies and is therefore easy to integrate into web applications. Web principles like URIs and links can be applied to data that is published in RDF format.

As soon as a local or global data source is established, it is possible to use inferencing and reasoning to generate new data. This is a very important advantage and a key element for the Semantic Web. New data (that was previously unknown) can be generated out of several connected data sources. Data does not have to be complete in the first place if missing elements can be inferred from the incomplete datasets.

With these options in mind, the vacation search example mentioned in the introduction chapter of this thesis seems much more realistic.

Disadvantages of RDF

RDF has some shortcomings or disadvantages as well. In practical applications, using RDF often means introducing a lot of redundancy. This limits processing speed of applications. Data stored in relational databases can be retrieved much faster than data stored in RDF files (usually within a so-called ”triplestore”). Scalability of these triplestores is much lower than that of modern data warehousing systems or relational database management systems. This problem is because of that the Internet users are very accustomed to high retrieval speeds when it comes to information query. For real-time-applications or time critical systems that use a large dataset, a data format like RDF is not really an option.

This becomes even more of a problem when global data sources are connected. It is almost impossible to infer new information out of several large interconnected RDF data sources distributed over the web in any reasonable time. However, caching might be a solution for this problem since search engines also cache a huge amount of data in a very specific and specialized internal structure to retrieve results quickly.

Alternatives to RDF

RDF can be seen as relatively bulky and difficult to generate (for humans). As of today, several alternatives to annotating data using RDF files exist. A very important development over the last years was the introduction of microformats. Microformats can be used to directly annotate information within HTML-code. The annotation is very simple and does not use a triple-based structure. In many cases, the information is just embedded by using HTML ”class”-Attributes. An example for contact information represented using the hCard (short for HTML vCard) microformat might look like this:

```
<ul class="vcard">
  <li class="fn">Christian Weiss</li>
  <li class="org">Vienna University of Technology</li>
  <li class="tel">123-456-789</li>
</ul>
```

Here, the formatted name (fn), organization (org) and telephone number (tel) have been identified using specific class names defined in the hCard format and the contact information is wrapped up in the class=”vcard”. Search engines or other applications can now use this information to extract and interpret the metadata.

Another advantage against using RDF-files is that the information is not duplicated (HTML and RDF files exist simultaneously) but just marked up within the original file. Regarding the realization of the Semantic Web Vision, microformats could be considered as a strong alternative because their simplicity can reduce the entry barrier to publish metadata on the web.

A combination of Microformats with RDF is possible by using RDFa [42]. RDFa enables metadata markup that is compatible with Semantic Web principles within HTML pages. Passages and words used on a website can be associated with semantic markup. RDFa can be seen as a connector between RDF and microformats.

RDFa and microformats are already used on the web. Search engine operator Google conducted some research in 2010 where they used a random sample of 1 million websites to count occurrences of these formats. A total of 40,000 pages out of the sample used microformats, while about 2,500 of them were applying RDFa to annotate information. [50]. A more recent analysis from 2012 conducted by Yahoo [29] uses a much larger sample of more than 3 billion records. They authors come to the conclusion that the adoption of markup on the web has improved and that more than 30% of web pages contain some form of semantic markup. In comparison to previous studies they show a significant growth of RDFa markup that has clearly outnumbered microformats by the time the study was conducted. Out of the dataset used 795 million pages contained RDFa markup while 272 million used microformats. However, it has to be noted that the growth of RDFa pages is mainly attributed to facebook.com using RDFa as a basis for their Open Graph Protocol [40] which accounts for 711 million pages.

Microformats and RDFa are (in most cases) open standards. However, companies also try to establish proprietary formats for semantic annotation. E.g. the Google Merchant Center [28] tries to establish specific formats and vocabulary for annotating product data in the E-Commerce industry. Schema.org [44] is a more open approach that was created by search engine providers like Microsoft (Bing), Google and Yahoo. The project hosts several metadata schemata and is an attempt to define a shared vocabulary focused on popular concepts.

2.4 Ontologies

The term ”ontology” is not only related to computer science but describes the ”philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations” [35]. The term dates back to ancient Greek philosophy.

In computer science, an ontology is a formal specification of vocabularies, hierarchies and relations between concepts. When it comes to the Semantic Web, ontologies are used to model domain knowledge. This is a highly essential requirement for interconnected data sources and reasoning. Ontologies are used as a schema for everything else, a Semantic Web Vision cannot be realized without them.

In its simplest form, an ontology could just define a vocabulary or a hierarchy. This is called a taxonomy. The following example shows such a taxonomy for places or venues within a city:

```
Place
  Leisure
    Sightseeing
      Museum
      Park
    Sports venue
      Stadium
      Swimming pool
  Educational Institution
    School
    University
  Health Care
    Hospital
    Defibrillator
```

By using an ontology that defines this hierarchy it can already be reasoned, that a Swimming pool is a Sports venue and School is not. If there is an RDF triple that states that a certain individual (identified via a URI) is of type Swimming pool, it can be automatically deduced that this is a Place (and, more specifically, also a Sports venue). When Swimming pool is seen as a class, ”Sports venue” would be its superclass and ”Leisure” would be the superclass of ”Sports venue”. In this example there is only one superclass per item. However there could be another definition:

```
Indoor Venue
  School
  Museum
Outdoor Venue
  Swimming pool
  Park
```

This definition assigns different superclasses to the already mentioned classes. This could be done by adding a second ontology but can also be stated in the same ontology since every class can have more than one superclass.

In addition to just defining a hierarchy, ontologies can also add properties to classes. For example, it would be possible to define attributes like name, color or calories to our Place classes.

There are several languages used to define ontologies.

RDF Schema

RDF Schema is a simple ontology language. It is standardized via a W3C recommendation [43]. RDF Schema uses RDF for notation and is comparable to what XML Schema is for XML. A simple example that defines two classes and states that one is a subclass (the inverse of a superclass) of the other one could look like this:

```
<rdfs:Class rdf:ID="Leisure"/>
<rdfs:Class rdf:ID="Swimming pool">
  <rdfs:subClassOf rdf:resource="#Leisure"/>
</rdfs:Class>
```

RDF offers the following vocabulary:

Resource All objects and items in RDF Schema are a resource. This is the superclass of all other classes.

Class All classes that are defined with RDF schema are an instance of class.

Literal Literals can hold textual or numerical values.

Property RDF properties use this type. A property is assigned to a class and defines its features.

Individual Individuals are specific instances of classes.

Important features of RDF and their meaning for reasoning purposes are:

Class defines a group of individuals that share characteristics. For example, the individuals ”Stadtpark” and ”Augarten” might both be of class ”Park”.

rdfs:subClassOf Classes can be organized in a hierarchy using rdfs:subClassOf.For example, the classes ”Swimming pool” and ”Stadium” can both be subclasses of ”Sports venue”. A reasoner can easily deduce that an instance of ”Stadium” is also an instance of ’Sports venue”.

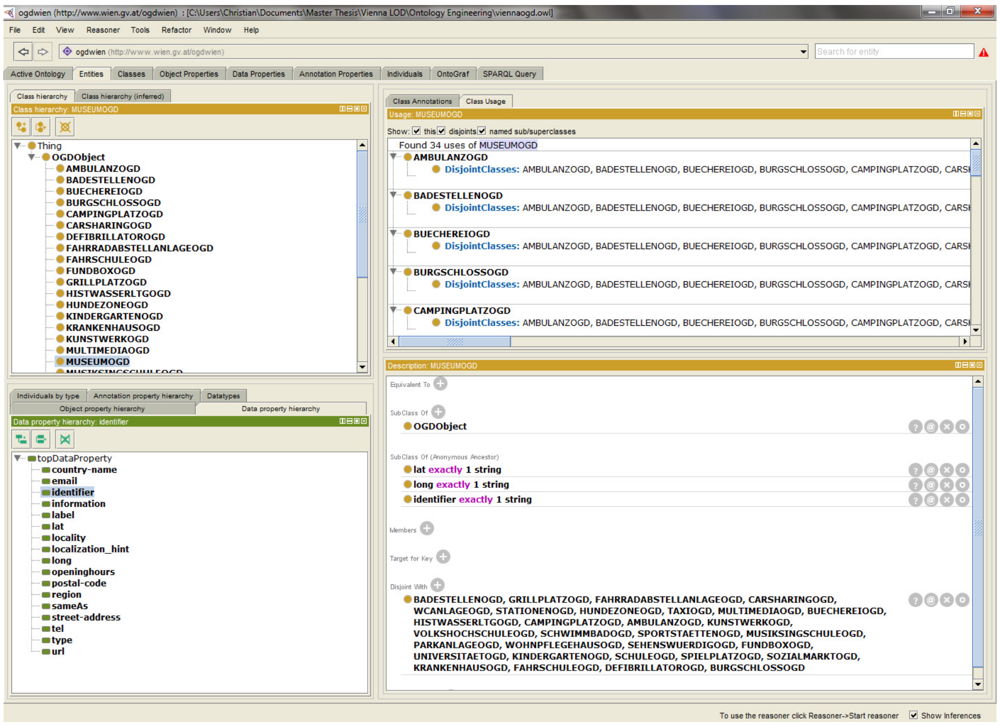
rdf:Property As already stated, properties get assigned to a class and define the features that certain individuals (instances) of the class have. This is done by assigning a literal to a class via a property (data property). Properties can also be used to state relationships between individuals by assigning another individual or resource via a property (object property).

rdfs:range By using rdfs:range, the individuals that any property may use as its value can be limited. E.g. the property ”isLocationType” can be limited to use only classes of type ”Location type” as its domain. A reasoner can easily deduce that something is assigned by using the object property ”isLocationType” is of class ”Location type”.

rdfs:domain By using rdfs:domain, it is possible to limit the individuals to which a certain property can be applied. E.g. the property ”isLocationType” can be limited to use only classes of type ”Place” as its domain. A reasoner can easily deduce that something that has the property ”isLocationType” assigned is of class ”Place”.

In general, RDF Schema enables the creation of simple classifications and taxonomies. It allows to define some vocabulary that is important to enable interconnected distributed data sources. Even some simple inferencing is possible using RDF Schema ontologies. For example, the Swimming pool inference example from before can be implemented by using just an RDF Schema ontology.

However, RDF Schema is not sufficient to describe more complex features and relations. It is not possible to express disjointness, e.g. to define that a Educational Institution cannot be an School and a University at the same time. There are no cardinality constraints (e.g. it is not possible to express that a cocktail must have at least two ingredients) or inverse properties.


ogdwien (http://www.wien.gv.at/ogdwien) : [C:\Users\Christian\Documents\Master Thesis\Vienna LOD\Ontology Engineering\viennaogd.owl]
File Edit View Reasoner Tools Refactor Window Help
ogdwien (http://www.wien.gv.at/ogdwien)
Search for entity
Active Ontology
Entities
Classes
Object Properties
Data Properties
Annotation Properties
Individuals
OntoGraf
SPARQL Query
Class hierarchy
Class hierarchy (inferred)
Class hierarchy: MUSEUMOGD
Individuals by type
Annotation property hierarchy
Datatypes
Object property hierarchy
Data property hierarchy
Data property hierarchy: identifier
Class Annotations
Class Usage
Usage: MUSEUMOGD
Show: this disjoints named sub/superclasses
Description: MUSEUMOGD
Equivalent To
SubClass Of
SubClass Of (Anonymous Ancestor)
Members
Target for Key
Disjoint With
To use the reasoner click Reasoner->Start reasoner
Show Inferences

```
Thing
  OGDObject
    AMBULANZOGD
    BADESTELLENOGD
    BUECHEREIOGD
    BURGSCHLOSSOGD
    CAMPINGPLATZOGD
    CARSHARINGOGD
    DEFIBRILLATOROGD
    FAHRRADABSTELLANLAGEOGD
    FAHRSCHULEOGD
    FUNDBOXOGD
    GRILLPLATZOGD
    HISTWASSERLTGOGD
    HUNDEZONEOGD
    KINDERGARTENOGD
    KRANKENHAUSOGD
    KUNSTWERKOGD
    MULTIMEDIAOGD
    MUSEUMOGD
```

```
topDataProperty
  country-name
  email
  identifier
  information
  label
  lat
  locality
  localization_hint
  long
  openinghours
  postal-code
  region
  sameAs
  street-address
  tel
  type
  url
```

```
Found 34 uses of MUSEUMOGD
AMBULANZOGD
  DisjointClasses: AMBULANZOGD, BADESTELLENOGD, BUECHEREIOGD, BURGSCHLOSSOGD, CAMPINGPLATZOGD, CARSI
BADESTELLENOGD
  DisjointClasses: AMBULANZOGD, BADESTELLENOGD, BUECHEREIOGD, BURGSCHLOSSOGD, CAMPINGPLATZOGD, CARSI
BUECHEREIOGD
  DisjointClasses: AMBULANZOGD, BADESTELLENOGD, BUECHEREIOGD, BURGSCHLOSSOGD, CAMPINGPLATZOGD, CARSI
BURGSCHLOSSOGD
  DisjointClasses: AMBULANZOGD, BADESTELLENOGD, BUECHEREIOGD, BURGSCHLOSSOGD, CAMPINGPLATZOGD, CARSI
CAMPINGPLATZOGD
  DisjointClasses: AMBULANZOGD, BADESTELLENOGD, BUECHEREIOGD, BURGSCHLOSSOGD, CAMPINGPLATZOGD, CARSI
```

```
SubClass Of
  OGDObject
SubClass Of (Anonymous Ancestor)
  lat exactly 1 string
  long exactly 1 string
  identifier exactly 1 string
Disjoint With
  BADESTELLENOGD, GRILLPLATZOGD, FAHRRADABSTELLANLAGEOGD, CARSHARINGOGD,
  WCANLAGEOGD, STATIONENOGD, HUNDEZONEOGD, TAXIOGD, MULTIMEDIAOGD, BUECHEREIOGD,
  HISTWASSERLTGOGD, CAMPINGPLATZOGD, AMBULANZOGD, KUNSTWERKOGD,
  VOLKSHOCHSCHULEOGD, SCHWIMMBADOGD, SPORTSTAETTENOGD, MUSIKSINGSCHULEOGD,
  PARKANLAGEOGD, WOHNPFLEGEHAUSOGD, SEHENSWUERDIGOGD, FUNDBOXOGD,
  UNIVERSITAETOGD, KINDERGARTENOGD, SCHULEOGD, SPIELPLATZOGD, SOZIALMARKTOGD,
  KRANKENHAUSOGD, FAHRSCHULEOGD, DEFIBRILLATOROGD, BURGSCHLOSSOGD
```

Figure 2.4: Editing an OWL ontology in Protege

Web Ontology Language (OWL)

The Web Ontology Language was introduced to overcome the restrictions of RDF Schema. Like RDFS it was released as a W3C recommendation [36]. OWL is built upon RDF Schema and introduces many extensions and features that enable the definition of complex ontologies.

OWL is very well supported by several tools and frameworks. The visual ontology editor Protege (seen in figure 2.4) is a very popular tool that allows to create complex OWL ontologies including inferencing.

OWL, like RDFS, is based on XML. OWL also uses the RDF syntax and is therefore easy to read for everyone with RDF experience. According to [36], when seen in contrast to RDFS, OWL adds vocabulary for describing properties and classes. Some examples are:

- relations between classes, e.g. disjointness
- cardinality
- equality
- richer typing of properties

- characteristics of properties like symmetry
- enumerated classes

OWL is compatible with RDFS and even uses some of its features, most notably Class, rdfs:subClassOf, rdf:Property, rdfs:range, rdfs:domain. These features are used in the same context as described in the RDFS section of this chapter.

OWL syntax example

This section describes a very basic OWL Lite ontology in OWL syntax. The ontology is a minimalistic subset from the Place example ontology described above. Namespace and prefix definitions are omitted from this example.

The first part declares the four classes Place, Leisure, Educational Institution and Location type:

```
<Declaration>
 <Class IRI="#Place"/>
</Declaration>
<Declaration>
 <Class IRI="#Leisure"/>
</Declaration>
<Declaration>
  <Class IRI="#Educational Institution"/>
</Declaration>
<Declaration>
 <Class IRI="#Location type"/>
</Declaration>
```

By using subClassOf, the Classes Leisure and Educational Institution are defined as subclasses of Place:

```
<SubClassOf>
 <Class IRI="#Leisure"/>
 <Class IRI="#Place"/>
</SubClassOf>
<SubClassOf>
 <Class IRI="#Educational Institution"/>
 <Class IRI="#Place"/>
</SubClassOf>
```

The property isLocationType is declared as an object property (a property linking two individuals):

```
<Declaration>
 <ObjectProperty IRI="#isLocationType"/>
</Declaration>
```

The property isLocationType is restricted to the Domain ”Place” and the Range ”Location type”. Any instance of type ”Place” (and of course of ”Leisure” and ”Educational Institution” can now have a property defining its Location type by using any instance of type ”Location type”:

```
<ObjectPropertyDomain>
 <ObjectProperty IRI="#isLocationType"/>
 <Class IRI="#Place"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
 <ObjectProperty IRI="#isLocationType"/>
 <Class IRI="#Location type"/>
</ObjectPropertyRange>
```

OWL Sublanguages

Since there are many different use cases for ontologies on the web and - more specifically - different levels of expression, three sublanguages were defined. Those are OWL Lite, OWL DL (where the DL stands for description logic) and OWL Full. Each of the sublanguages can be seen as an extension of the former one. Therefore, every valid OWL Lite ontology is also a valid OWL Full ontology, however not the other way round. This relation can be seen in a diagram in figure 2.5.

The language overview listed at [36] features the following description of sublanguages:

OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite

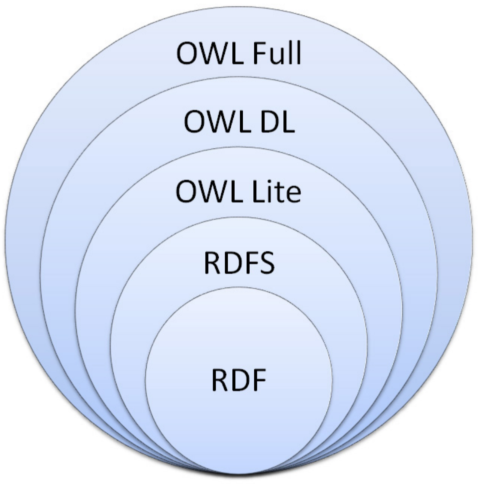


Figure 2.5: Relation between RDF, RDFS and OWL sublanguages

provides a quick migration path for thesauri and other taxonomies. OWL Lite also has a lower formal complexity than OWL DL.

OWL DL supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

When designing an ontology, especially when reasoning support is needed, it is very important to choose the right sublanguage for the use case at hand.

2.5 SPARQL

SPARQL (which is an acronym for SPARQL Protocol and RDF Query Language) is a query language for RDF data. SPARQL, like RDF and OWL, is a W3C Recommendation [47], officially introduced in 2008. Alternatives like RDQL or SeRQL, which are simple query languages for Semantic Web data, were already available before SPARQL but feature more restrictions. By using SPARQL it is possible to query RDF datasets similar to querying a relational database using SQL. Some elements of SPARQL seem related to such query languages for relational databases, however there are some important differences.

SPARQL supports four different query forms.

SELECT Similar to SQL, this query type returns raw results. The result usually has a table form (organized with variables as table header) but can also be delivered as XML or other formats.

CONSTRUCT The CONSTRUCT query type returns an RDF graph constructed by substituting variables in a set of triple templates. Therefore the result is valid RDF.

ASK Provides a simple true/false boolean-type result for a certain query. It is checked whether a pattern can be matched by the RDF data.

DESCRIBE This query type returns a valid RDF graph that describes resources. Therefore it is not required to already know the structure of RDF data when formulating the query.

The most widely used query form is SELECT. Using SELECT, it is possible to formulate simple informative queries.

SPARQL Syntax

Lets look at an example for a very simple query:

```
SELECT ?creatorname
WHERE
{
 <http://example.org/document2>
 <http://purl.org/dc/elements/1.1/creator>
 ?creatorname .
}
```

This query, if applied to the example from the RDF section of this chapter, returns the name of the creator of a certain document. URIs are used to reference subject (the document) and predicate (element ”creator” from the Dublin Code vocabulary). The object in this case is a variable, indicated by the ”?” prefix. The principle of SPARQL queries is easily visible in this example. The query defines RDF triples where some elements are replaced by variables. The query processor then tries to retrieve all values that would complete the triple. For example, the following query would list all documents and their creators (if the Dublin Code creator property is used).

```
SELECT ?document ?creatorname
WHERE
{
 ?document
 <http://purl.org/dc/elements/1.1/creator>
 ?creatorname .
}
```

It is possible to define namespace prefixes to write less cluttered query expressions. For this example the Dublin Core namespace could be defined by adding ”PREFIX dc: <http://purl.org/dc/elements/1.1/>” before the start of the select query. In this case the creator property can be referenced as ”dc:creator”. Adding namespace prefixes increases the readability of more complex queries. A further option is to define optional parts of a query. Resources that fulfill all mandatory parts but not the optional parts will be added to the result set as well. A more advanced query example could look like this:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?document ?creatorname ?title
WHERE
{
 ?document
 dc:creator
 ?creatorname .
OPTIONAL {
 ?document
 dc:title
 ?title .
}
```

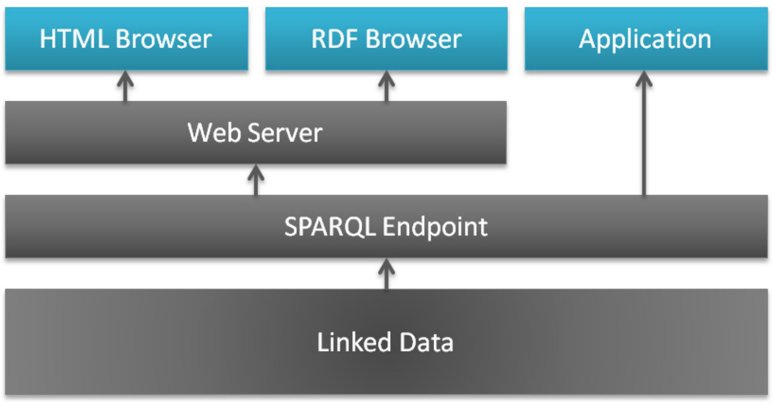


Figure 2.6: How data can be served using a SPARQL endpoint.

```
 FILTER (?creatorname = "Christian Weiss").
} ORDER BY ASC(?creatorname)
```

This example also shows other SPARQL concepts. A filter is applied to only allow results in which a variable has a certain value. It is also possible to apply numeric comparisons, e.g. to allow only results where a variable is higher or lower than a certain value. The ”ORDER BY” clause defines a sorting variable and order. In this case, the results are ordered by the variable ”?creatorname” in ascending order.

Using SPARQL

There is a number of Semantic Web frameworks that support querying with SPARQL. RDF data is usually maintained within triplestores where queries can be evaluated against a very large set of triples. However, data on the Semantic Web is supposed to be openly available. Therefore, many data sources are available via so-called SPARQL endpoints. A SPARQL endpoint enables users or machines to query a dataset remotely, similar to accessing a relational database over a remote connection. A SPARQL endpoint is a web service that can be accessed via a URL. For example, the already mentioned DBpedia has a public SPARQL endpoint available at ”http://dbpedia.org/sparql”. An easy way to send queries to such an endpoint is by using a framework or library like Apache Jena [20]. SPARQL can also be accessed by a web server to serve RDF or HTML data to browsers (e.g. resolving of URIs). In this case a layer translates incoming requests to SPARQL queries and returns the result (usually a describe query about the URI requested). The principle is shown in figure 2.6.

CHAPTER 3

State of the Art in Linked Open Data

3.1 Linked Open Data: Idea and Principles

The idea of Linked Open Data (LOD) emerged in 2007, when the Linking Open Data project [39] was started. Linked Open Data stands for data published on the web in an open, machine-readable format that is interlinked with other data sources. Existence of LOD is a very important requirement for the realization of a global Semantic Web vision.

In 2008, a paper was released by Christian Bizer et al. that states [5]:

> The web is increasingly understood as a global information space consisting not just of linked documents, but also of Linked Data. More than just a vision, the resulting web of data has been brought into being by the maturing of the Semantic Web technology stack, and by the publication of an increasing number of datasets according to the principles of Linked Data.

This paper describes how linked data can be created using the RDF standard: ”The glue that holds together the traditional document Web is the hypertext links between HTML pages. The glue of the data web is RDF links.” [5].

A definition of linked data is published in [4]:

> The term Linked Data refers to a set of best practices for publishing and connecting structured data on the web. These best practices have been adopted by an increasing number of data providers over the last three years, leading to the creation of a global data space containing billions of assertions - the web of data.

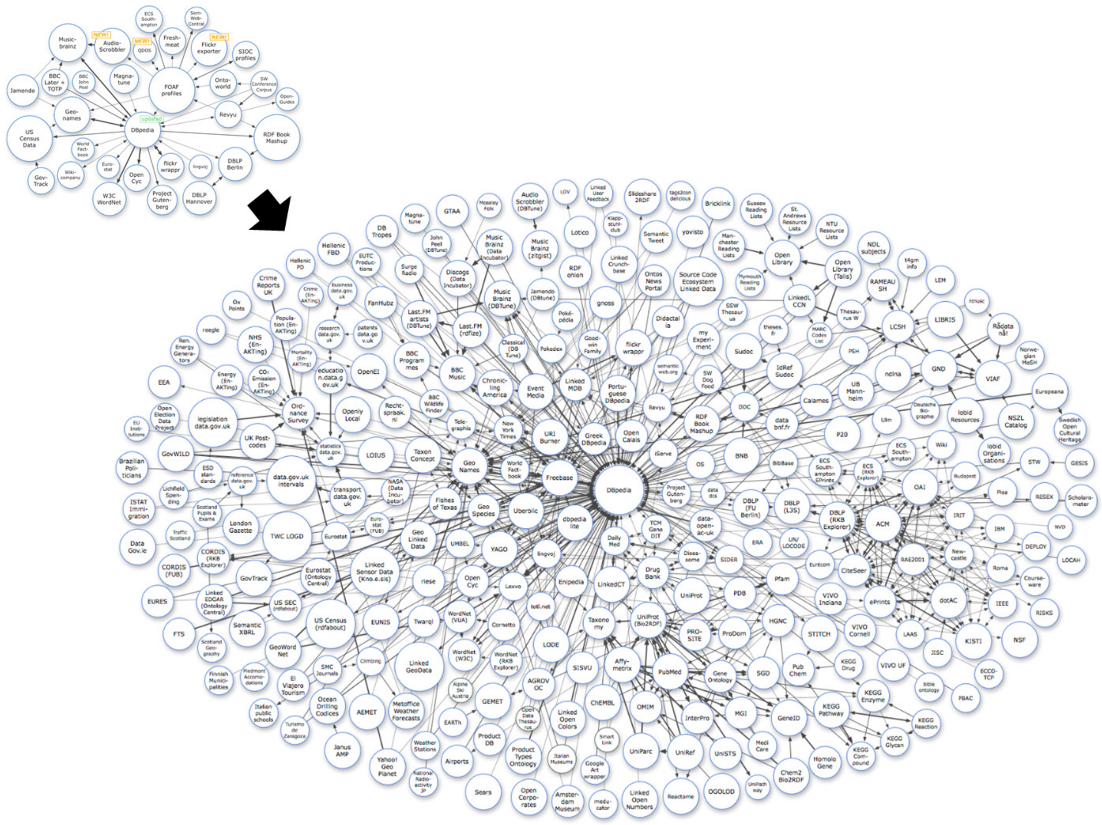


Figure 3.1: Expansion of the LOD cloud between 2008 and late 2011 (Source: [10])

3.2 The Linked Open Data cloud

There was a massive growth of data published as LOD during the last years. This development is visible in the LOD cloud [10], a diagram of linked data on the web. It has grown from a small set of data sources into a highly interlinked and very extensive graph. The development is depicted in figure 3.1. The growth of linked data is also visible in a statistic report from the Data Hub [49], a data management platform that enables content providers to register their datasets. These datasets can then be searched and discovered using the Data Hub website. Figure 3.2 shows the increasing number of LOD datasets registered on the portal.

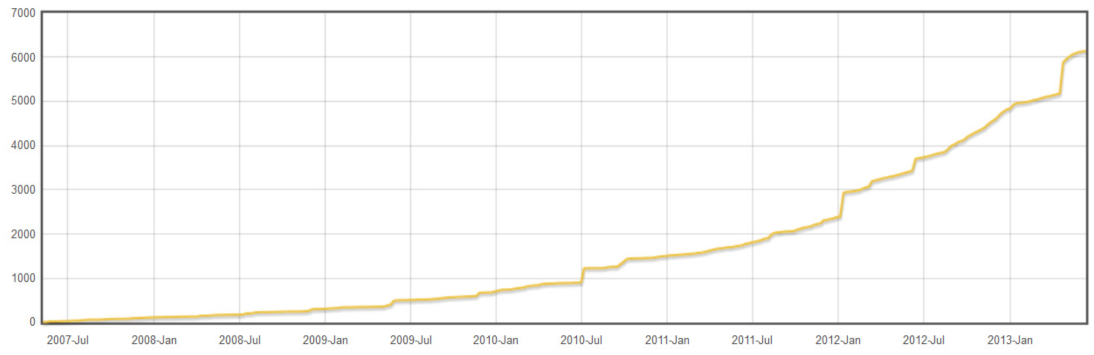


Figure 3.2: Growth of LOD datasets on the Data Hub (Source: [49])

3.3 Linked Open Data Technologies

Linked Open Data is usually based on the Semantic Web technologies that were described in chapter 2. There is no perfect definition whether a published dataset qualifies as Linked Open Data or not, instead several rules and guidelines exist that state requirements for linked data on the web exist. In many cases, linked data on the web is published in several formats in parallel (e.g. DBpedia [12]).

Tim Davies created a stack of LOD technologies [48] that is similar to the Semantic Web layer cake described in chapter 2. It is depicted in figure 3.3.

This LOD stack is a good starting point for understanding how LOD should be set up since it shows the technologies that are required and their relation. The World Wide Web (HTTP) serves as a basis for serving LOD documents. Since HTTP is used, URLs (URIs) define locations and resources. RDF is used to exchange data and ontologies are utilized to describe schemata. Contrary to the Semantic Web layer cake released earlier, XML is not a requirement and it is explicitly mentioned that there are other ways to serialize RDF data. Several vocabularies already exist on the web (often in form of an ontology) that should be re-used when creating Linked Open Data. The diagram also addresses the issue of storing data, where flat RDF files are mentioned alongside triplestores. However, querying data via SPARQL usually requires the existence of a triplestore featuring a SPARQL endpoint.

Since the LOD stack is only about data, the layers about logic, proof and trust are absent. A ”License” layer is introduced instead that highlights the importance of attaching an open license to LOD. Finally an application layer exists that introduces usage scenarios.

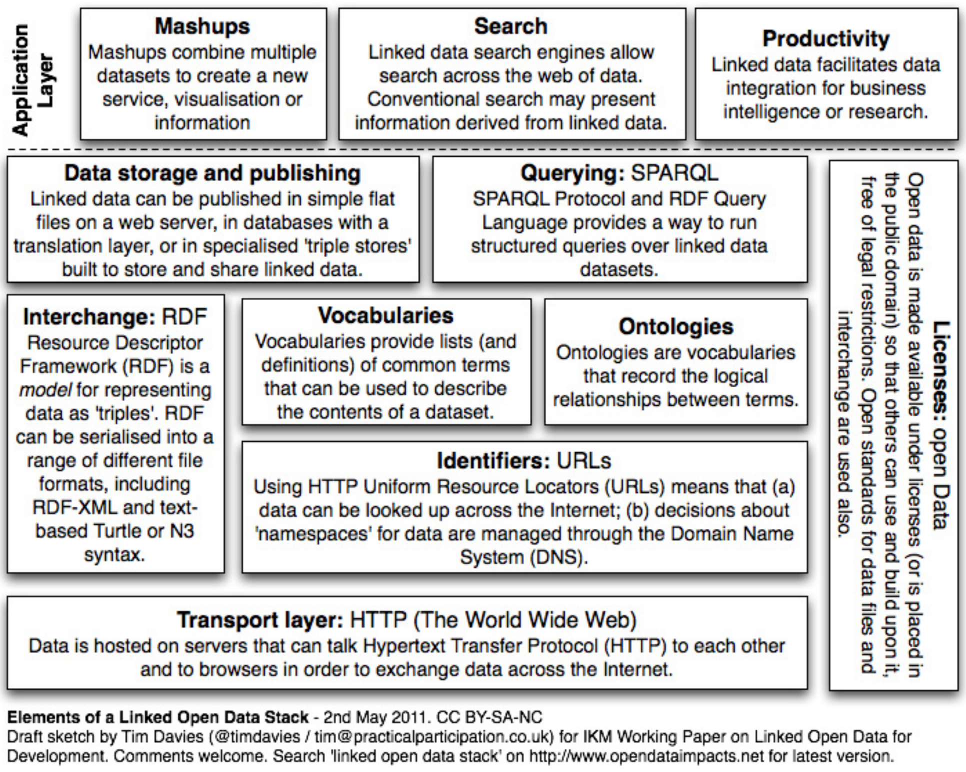

Application Layer
Mashups
Mashups combine multiple datasets to create a new service, visualisation or information
Search
Linked data search engines allow search across the web of data. Conventional search may present information derived from linked data.
Productivity
Linked data facilitates data integration for business intelligence or research.
Data storage and publishing
Linked data can be published in simple flat files on a web server, in databases with a translation layer, or in specialised 'triple stores' built to store and share linked data.
Querying: SPARQL
SPARQL Protocol and RDF Query Language provides a way to run structured queries over linked data datasets.
Licenses: open Data
Open data is made available under licenses (or is placed in the public domain) so that others can use and build upon it, free of legal restrictions. Open standards for data files and interchange are used also.
Interchange: RDF
Resource Descriptor Framework (RDF) is a model for representing data as 'triples'. RDF can be serialised into a range of different file formats, including RDF-XML and text-based Turtle or N3 syntax.
Vocabularies
Vocabularies provide lists (and definitions) of common terms that can be used to describe the contents of a dataset.
Ontologies
Ontologies are vocabularies that record the logical relationships between terms.
Identifiers: URLs
Using HTTP Uniform Resource Locators (URLs) means that (a) data can be looked up across the Internet; (b) decisions about 'namespaces' for data are managed through the Domain Name System (DNS).
Transport layer: HTTP (The World Wide Web)
Data is hosted on servers that can talk Hypertext Transfer Protocol (HTTP) to each other and to browsers in order to exchange data across the Internet.
Elements of a Linked Open Data Stack - 2nd May 2011. CC BY-SA-NC
Draft sketch by Tim Davies (@timdavies / tim@practicalparticipation.co.uk) for IKM Working Paper on Linked Open Data for Development. Comments welcome. Search 'linked open data stack' on http://www.opendataimpacts.net for latest version.

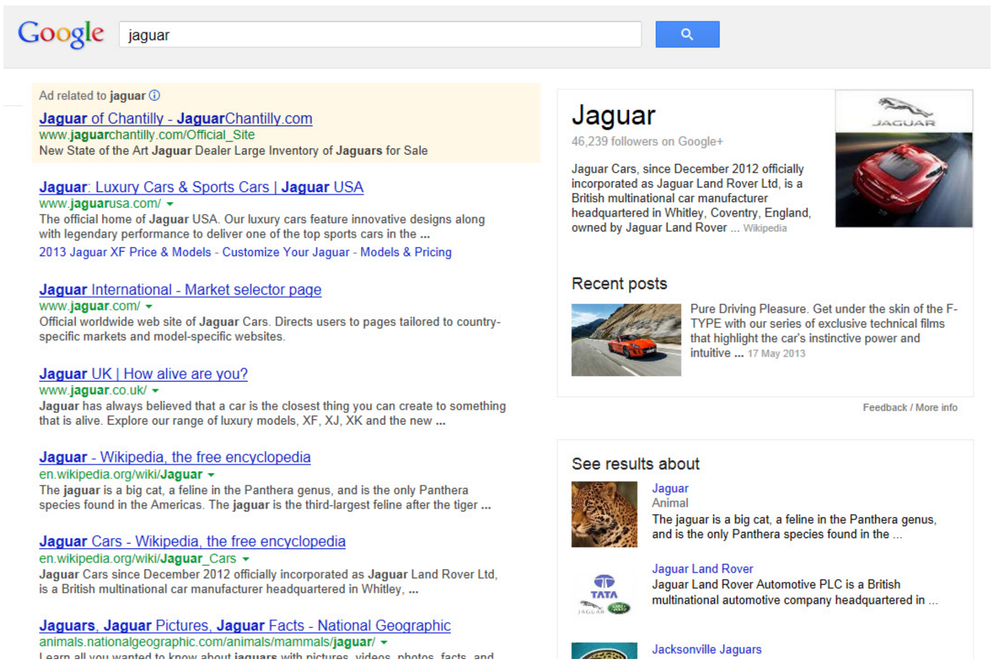
Figure 3.3: A stack of LOD technologies (Source: [48])

3.4 Linked Open Data Scenarios

Mashups

Mashups combine several data sources to create a consolidated dataset. The resulting combined data is more useful than the isolated data sources on their own. An example for a mashup would be an online map that is combined with a picture database. If geographic positions of pictures are known they can be displayed on the map. When the Web 2.0 emerged and many portals featuring user-generated data were created, mashups became increasingly important. Of course data quality is especially important for creating mashups since datasets have to be matched with high precision to create useful data.

LOD is especially important for mashups because unlike mashups that are designed using fixed datasets, LOD-enabled mashups operate using linked, open and dynamic datasets. These mashups than deliver more accurate and up-to-date answers and solutions to queries. Availability of LOD also enables easier and faster creation of mashups since data can be accessed in a unified way while static mashups often have to handle a variety of data formats.



jaguar

Ad related to **jaguar**

Jaguar of Chantilly - **Jaguar**Chantilly.com
www.**jaguar**chantilly.com/Official_Site
New State of the Art **Jaguar** Dealer Large Inventory of **Jaguars** for Sale

Jaguar: Luxury Cars & Sports Cars | **Jaguar** USA
www.**jaguar**usa.com/
The official home of **Jaguar** USA. Our luxury cars feature innovative designs along with legendary performance to deliver one of the top sports cars in the ...
2013 Jaguar XF Price & Models - Customize Your Jaguar - Models & Pricing

Jaguar International - Market selector page
www.**jaguar**.com/
Official worldwide web site of **Jaguar** Cars. Directs users to pages tailored to country-specific markets and model-specific websites.

Jaguar UK | How alive are you?
www.**jaguar**.co.uk/
Jaguar has always believed that a car is the closest thing you can create to something that is alive. Explore our range of luxury models, XF, XJ, XK and the new ...

Jaguar - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Jaguar**
The **jaguar** is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the Americas. The **jaguar** is the third-largest feline after the tiger ...

Jaguar Cars - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Jaguar**_Cars
Jaguar Cars since December 2012 officially incorporated as **Jaguar** Land Rover Ltd, is a British multinational car manufacturer headquartered in Whitley, ...

Jaguars, **Jaguar** Pictures, **Jaguar** Facts - National Geographic
animals.nationalgeographic.com/animals/mammals/**jaguar**/

Jaguar
46,239 followers on Google+
Jaguar Cars, since December 2012 officially incorporated as Jaguar Land Rover Ltd, is a British multinational car manufacturer headquartered in Whitley, Coventry, England, owned by Jaguar Land Rover ... Wikipedia

Recent posts
Pure Driving Pleasure. Get under the skin of the F-TYPE with our series of exclusive technical films that highlight the car's instinctive power and intuitive ... 17 May 2013

Feedback / More info

See results about
Jaguar
Animal
The jaguar is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the ...
Jaguar Land Rover
Jaguar Land Rover Automotive PLC is a British multinational automotive company headquartered in ...
Jacksonville Jaguars

Figure 3.4: A Google search featuring some semantics from linked data

Search

Traditional search engines on the web look for the occurrence of search terms within websites or in links between websites. They do not semantically understand what the user is looking for and just do string comparison. A Semantic Web enabled search engine that is able to semantically process user questions would in theory deliver results with much higher precision. However, aside from availability of Linked Open Data the realization of such search engines also needs working concepts and solutions for the proof and trust layers of the Semantic Web. Anyway, as of today search engines already use linked data to some extent. Figure 3.4 shows a Google search featuring semantics. The search for ”Jaguar” not only returns a box describing the company Jaguar but another box that allows users to state whether they are looking for the animal Jaguar, the car company Jaguar or a football team with the same name. There are several ways Google is already using linked data to improve search results [50].

3.5 Linked Open Data: Design Guidelines

Several guidelines exist for the creation of Linked Open Data. Published back in 2006, a short article by Tim Berners-Lee [19] describes the basic requirements. To begin with, this article states four simple rules which are:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs. so that they can discover more things.

Basically, following the Semantic Web principles outlined in chapter 2 and using the technologies associated with it already guarantees to generate valuable open data. Anyway, links to other data sources need to be introduced to generate Linked Open Data.

LOD Star scheme

The article [19] concludes with introducing a star scheme for rating the quality of data on the web that reads:

1 Star Available on the web (whatever format) but with an open license, to be Open Data

2 Stars Available as machine-readable structured data (e.g. excel instead of image scan of a table)

3 Stars as (2) plus non-proprietary format (e.g. CSV instead of excel)

4 Stars All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff

5 Stars All the above, plus: Link your data to other people's data to provide context

URI Design Issues

Using URIs is a very basic requirement for creating Linked Open Data. Anyway, it is also important to use meaningful URIs. The W3C Linked Open Data Cookbook [8] introduces more requirements and guidelines for creating such URIs. The document also states that URIs should be in HTTP format but also urges to use clean and stable URIs and encourages developers to create URIs only for domains they control.

In addition, developers are encouraged to use natural keys like ”building357” instead of keys like ”b2357”. It is not recommended to include version numbers or technology names within URIs. The reason is that technologies might change but the identifiers should stay the same. It is also difficult to link to resources that are versioned because then the resource referenced will soon be outdated. URIs that don't include such identifiers are called ”neutral URIs”. The same problem occurs for using dates within URIs, something that is also discouraged in the W3C document. Only in some cases using dates in URIs actually makes sense. An example are W3C recommendations. URIs of these documents include the date of publication. The Linked Open Data cookbook explains these cases:

> The W3C is a well known exception to this thumb rule. They use a convention for URIs related to the dates that working groups are established or documents are standardized. For example, the W3C RDF Recommendation was published 10 February 2004, so the convention they use is to path with the date as follows http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/. This approach is acceptable however, it should be used with careful consideration on how things may change over time.

3.6 Linked Open Data: Existing Vocabulary

When creating properties for RDF data it is recommended to use vocabulary that is already defined and used in other data sources. This makes it much easier for machines (i.e. reasoning engines) to interpret the structure of the published data. Vocabulary for many use cases is already defined in publicly available ontologies and widely used over several important Linked Open Data sources. It is very easy to re-use vocabulary by just defining a prefix, listing the namespace and using the elements defined there. The following list describes options for vocabulary that can be used when creating LOD.

RDFS and OWL The default vocabulary of RDFS [43] and OWL [36] already contains elements that are well-known and useful when creating linked data. The element rdfs:label is often used to label resources. This label can later be used when displaying data to users. Another important element is owl:sameAs. This element can be used to define similarity between resources.

vCard The vCard format was designed for defining electronic business cards. It is widely supported by E-Mail clients and often used to transmit contact data between people or organizations. Since this format is very popular, an RDF encoding [53] exists that can be used when creating LOD. This RDF implementation defines properties such as ”postal-code”, ”street-address” or ”email” that can be used to describe addresses or contact data.

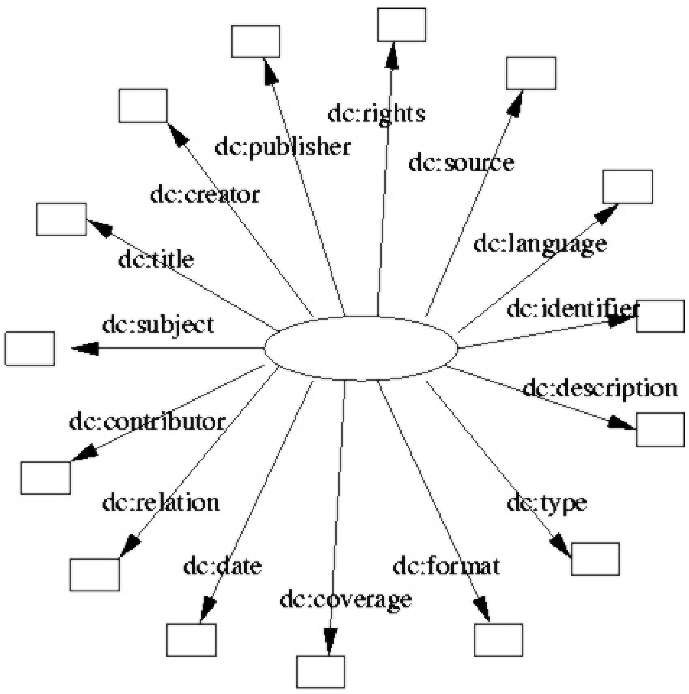


Figure 3.5: An illustration of properties defined by the Dublin Core vocabulary (Source: [18])

FOAF The Friend-of-a-Friend (FOAF) vocabulary defines terms to describe people, their activities and their relations to other people. When the Semantic Web started to emerge a lot of ontologies were created but not many were widely used. The FOAF ontology then became one of the first widely used ontologies [14]. FOAF defines an RDF/XML vocabulary to define personal information like name, homepage, mailbox and relations to friends and other properties. The property foaf:name is sometimes used as an alternative to rdfs:label for labeling resources.

Dublin Core Dublin Core is a metadata initiative [18] that provides properties that are suitable for describing documents. Attributes like title, creator, date, subject or publisher are available. Dublin Core has become a very popular vocabulary. Figure 3.5 shows the properties that are available.

WGS84 The WGS84 ontology defines terms for lat(itude), long(itude) and other information about spatially-located things, using WGS84 as a reference datum. [8]

BIBO The Bibliographic Ontology (BIBO) provides main concepts and properties for describing citations and bibliographic references (i.e. quotes, books, articles, etc). [8]

3.7 The DBpedia Example

DBpedia [12] is one of the most popular Linked Open Data sources. It can be seen as a best-practice example and therefore will be described in detail.

DBpedia is a community project that tries to build a large LOD database by extracting structured information from Wikipedia, a community-authored free online encyclopedia. The idea is similar to this thesis project - taking data that is somehow structured but does not use LOD vocabulary and transform it to a dataset that follows Linked Open Data guidelines by re-using vocabulary and publishing in Semantic Web formats. According to [2], Wikipedia is available in over 250 languages, with the English edition alone accounting for more than 1.95 million articles as of 2007. The DBpedia project now describes 3.77 million things, out of which 2.35 million are classified in a consistent ontology [12]. The full dataset consist of 1.89 billions of RDF triples and localized versions are available in more than 100 languages.

Targets of DBpedia

In an article [6] the initiators of the project define the following contributions as targets of DBpedia:

> - We develop an information extraction framework that converts Wikipedia content into a rich multi-domain knowledge base. By accessing the Wikipedia live article update feed, the DBpedia knowledge base timely reflects the actual state of Wikipedia. A mapping from Wikipedia infobox templates to an ontology increases the data quality.
> - We define a Web-dereferenceable identifier for each DBpedia entity. This helps to overcome the problem of missing entity identifiers that has hindered the development of the web of data so far and lays the foundation for interlinking data sources on the Web.
> - We publish RDF links pointing from DBpedia into other Web data sources and support data publishers in setting links from their data sources to DBpedia. This has resulted in the emergence of a web of data around DBpedia.

The DBpedia project sees itself as a ”Nucleus for the Web of Data” [2]. It’s main advantage is the possibility to get results for highly sophisticated queries via SPARQL. DBpedia also includes a lot of classification data and therefore can be utilized to provide vocabulary and taxonomy for other linked datasets.

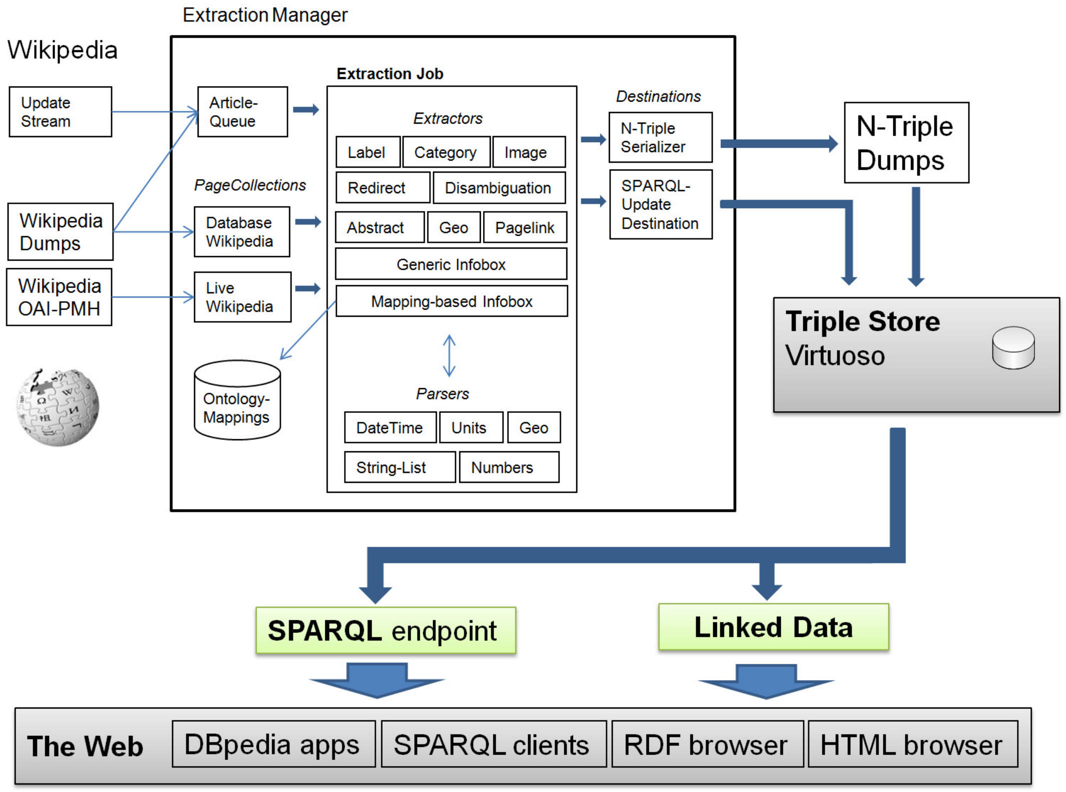


Figure 3.6: The architecture of DBpedia (Source: [6])

Conversion from Wikipedia to DBpedia

An introduction into how the DBpedia database is built can be found at [6]. This architecture is depicted in figure 3.6. The system uses Wikipedia SQL dumps that are published every month to gather data. The data is extracted mainly from categorizations and infoboxes that many articles feature. Infoboxes list the most important facts from an article. DBpedia uses attribute-value pairs from that infoboxes that can much more easily get extracted than continuous text. However, infoboxes often don't follow a common schema. At best, there are many schemata around and different communities use different schemata to describe the same kind of entity. DBpedia handles this by following 2 distinct approaches in parallel:

First, a generic approach is followed where all attributes that appear in an infobox are automatically transferred into statements where the property is auto-generated following the schema "http://dbpedia.org/property/" + name of the attribute. Some post-processing is applied to values. This generic approach guarantees complete coverage of data stored in infoboxes, however the data might not be suitable for queries covering a large set of data because of varying schemata (and therefore properties) used.

To solve the problem of different schemata used in infoboxes a mapping-based approach is followed as well. The most commonly used infobox attributes were added to an ontology and manually arranged. When data is converted using this approach even various units of measurement are normalized to create a high-quality dataset for complex queries to work with.

Conversion from Wikipedia to DBpedia: Content

According to [6] the following content is utilized for the conversion into Linked Open Data:

Labels Wikipedia articles always feature a title that is directly converted into an rdfs:label data property.

Categories Wikipedia articles are arranged in categories based on the SKOS vocabulary. SKOS specifications are W3C recommendations [51]. SKOS also provides the properties skos:concepts and skos:broader that are used by DBpedia.

Abstracts The first paragraph of an article is used as an abstract. It is directly converted info an rdfs:comment data property. A longer abstract is converted into a dbpedia:abstract property.

Interlanguage links Links that connect articles about the same topic in different language editions of Wikipedia are used to create labels in different languages.

Images Links pointing to images (hosted on Wikimedia Commons) are converted into foaf:depiction properties.

Redirects Wikipedia articles can redirect to other articles. References between DBpedia resources are created from this links.

Disambiguation Disambiguation links are converted into a dbpedia:disambiguates property.

External links DBpedia introduced a property dbpedia:reference for references to external web resources.

Pagelinks DBpedia introduced the property dbpedia:wikilink for internal references.

Homepages The foaf:homepage property from the Friend-of-a-Friend vocabulary is used to store links to homepages of organizations or entities.

Geo-coordinates Geo coordinates are also extracted and represented using the WGS84 vocabulary (lat and long properties).

Some of these approaches can be seen as best-practice patterns that can be used as an example for the conversion of the Vienna OGD datasets.

Accessing DBpedia data via SPARQL

There are various ways to access the DBpedia dataset. A SPARQL endpoint is available that can be remotely accessed at http://dbpedia.org/sparql. The endpoint supports some extensions to SPARQL like an option for full text search over selected properties. DBpedia allows to get answers to more complex questions. For example the following query lists actors that were starring in a movie directed by Christopher Nolan and how often they did so according to the current DBpedia dataset:

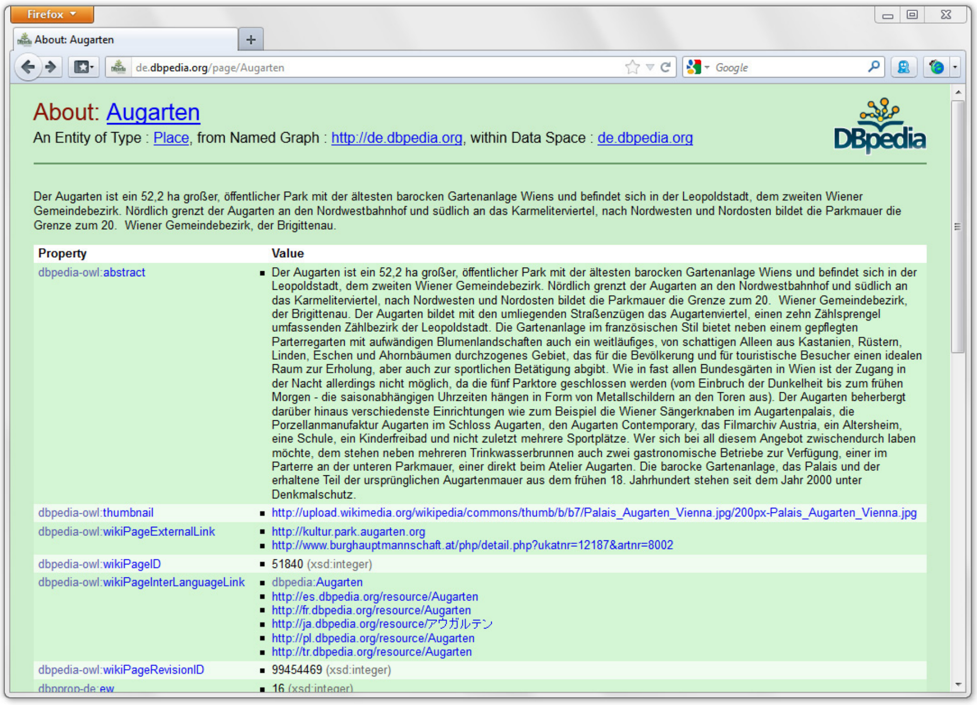
```
PREFIX : <http://dbpedia.org/resource/>
SELECT ?actor (count( ?actor) as ?countm)
WHERE {
 ?movie a dbpedia-owl:Film ;
 dbpedia-owl:starring ?actor;
 dbpedia-owl:abstract ?abstract;
 dbpedia-owl:director ?director.
 FILTER(LANGMATCHES(LANG(?abstract), "en")).
 FILTER (?director = :Christopher_Nolan).
}
ORDER BY DESC (?countm)
```

The DBpedia SPARQL endpoint will be used later to enrich the dataset resulting from the converted Vienna OGD.

Other options to access DBpedia data

It is also possible to browse the DBpedia data using a web browser (see figure 3.7). Data can be returned in various formats, including HTML, CSV and of course RDF in N-Triples, N3/Turtle, JSON or XML representation. DBpedia uses content negotiation to automatically return the correct format (e.g. HTML for web browsers, RDF for Semantic Web agents).

Another option is to acquire an RDF dump from DBpedia. The RDF dump contains even more data than that available through the SPARQL endpoint since infobox properties from further language editions are included. If this kind of localized information is important for an application, using the RDF dump is the only option to work with the data.



About: Augarten

An Entity of Type : Place, from Named Graph : http://de.dbpedia.org, within Data Space : de.dbpedia.org

Der Augarten ist ein 52,2 ha großer, öffentlicher Park mit der ältesten barocken Gartenanlage Wiens und befindet sich in der Leopoldstadt, dem zweiten Wiener Gemeindebezirk. Nördlich grenzt der Augarten an den Nordwestbahnhof und südlich an das Karmeliterviertel, nach Nordwesten und Nordosten bildet die Parkmauer die Grenze zum 20. Wiener Gemeindebezirk, der Brigittenau.

Property	Value
dbpedia-owl:abstract	▪ Der Augarten ist ein 52,2 ha großer, öffentlicher Park mit der ältesten barocken Gartenanlage Wiens und befindet sich in der Leopoldstadt, dem zweiten Wiener Gemeindebezirk. Nördlich grenzt der Augarten an den Nordwestbahnhof und südlich an das Karmeliterviertel, nach Nordwesten und Nordosten bildet die Parkmauer die Grenze zum 20. Wiener Gemeindebezirk, der Brigittenau. Der Augarten bildet mit den umliegenden Straßenzügen das Augartenviertel, einen zehn Zählsprengel umfassenden Zählbezirk der Leopoldstadt. Die Gartenanlage im französischen Stil bietet neben einem gepflegten Parterregarten mit aufwändigen Blumenlandschaften auch ein weitläufiges, von schattigen Alleen aus Kastanien, Rüstern, Linden, Eschen und Ahornbäumen durchzogenes Gebiet, das für die Bevölkerung und für touristische Besucher einen idealen Raum zur Erholung, aber auch zur sportlichen Betätigung abgibt. Wie in fast allen Bundesgärten in Wien ist der Zugang in der Nacht allerdings nicht möglich, da die fünf Parktore geschlossen werden (vom Einbruch der Dunkelheit bis zum frühen Morgen - die saisonabhängigen Uhrzeiten hängen in Form von Metallschildern an den Toren aus). Der Augarten beherbergt darüber hinaus verschiedenste Einrichtungen wie zum Beispiel die Wiener Sängerknaben im Augartenpalais, die Porzellanmanufaktur Augarten im Schloss Augarten, den Augarten Contemporary, das Filmarchiv Austria, ein Altersheim, eine Schule, ein Kinderfreibad und nicht zuletzt mehrere Sportplätze. Wer sich bei all diesem Angebot zwischendurch laben möchte, dem stehen neben mehreren Trinkwasserbrunnen auch zwei gastronomische Betriebe zur Verfügung, einer im Parterre an der unteren Parkmauer, einer direkt beim Atelier Augarten. Die barocke Gartenanlage, das Palais und der erhaltene Teil der ursprünglichen Augartenmauer aus dem frühen 18. Jahrhundert stehen seit dem Jahr 2000 unter Denkmalschutz.
dbpedia-owl:thumbnail	▪ http://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/Palais_Augarten_Vienna.jpg/200px-Palais_Augarten_Vienna.jpg
dbpedia-owl:wikiPageExternalLink	▪ http://kultur.park.augarten.org ▪ http://www.burghauptmannschaft.at/php/detail.php?ukatnr=12187&artnr=8002
dbpedia-owl:wikiPageID	▪ 51840 (xsd:integer)
dbpedia-owl:wikiPageInterLanguageLink	▪ dbpedia:Augarten ▪ http://es.dbpedia.org/resource/Augarten ▪ http://fr.dbpedia.org/resource/Augarten ▪ http://ja.dbpedia.org/resource/アウガルテン ▪ http://pl.dbpedia.org/resource/Augarten ▪ http://tr.dbpedia.org/resource/Augarten
dbpedia-owl:wikiPageRevisionID	▪ 99454469 (xsd:integer)
dbpprop-de:ew	▪ 16 (xsd:integer)

Figure 3.7: A DBpedia resource displayed in a web browser

DBpedia: Impact and Outlook

DBpedia has become one of the central hubs for the web of data. The LOD cloud diagram and the underlying source data [10] clearly show that DBpedia is the dataset that has the highest number of links (mostly incoming). The data that is included in the DBpedia database consists of many items that are of interest for other datasets and can be used to describe real-world entities. In addition, DBpedia is highly relevant for classification of things in the web of data.

According to [6] there will be a more seamless integration of DBpedia into Wikipedia. A MediaWiki extension will allow Wikipedia editors to augment their articles with data from DBpedia while also introducing consistency checking to improve on the problem of concurrent schemata.

3.8 Linked Open Government Data

Linked Open Data was at first mostly provided from non-government initiatives (although public funding was involved in some cases). In the past, government data was often kept secret - or at least not accessible to the public. During the last years there was a paradigm shift in this area and more and more government agencies started publishing data online as a public service. The city government of Vienna is one of these agencies. However, in this case (and in many other cases) the published data was not designed to conform to LOD requirements. Some governments like the UK have, however, already started to release data in the form of Linked Open Data. It can be seen from the LOD cloud that a relatively large part of LOD released on the web is actually governmental data. However, these data sets are not as interlinked as other datasets like publications or life sciences. A reason for this might be that governments only recently started to release OGD and there not much uniformity was established yet.

3.9 Advantages of Open Government Data

A study released in 2010 by Lucke and Geiger lists and discusses a variety of reasons why publishing government data online provides benefit and value to government agencies [54]:

1. Governments are becoming more open

   The introduction of open data published by governments follows public demand for higher transparency, increased accessibility and focus on citizens. An informed society becomes a stronger society that allows citizens to make informed judgments and participate in public decisions. Availability of open data from governments would therefore strengthen democracy. Discrimination of certain groups is avoided when data is available to everyone and everyone is able to use this data. Open data is also very useful in means of supervision (of government agencies). In the end, the gathering and creation of public data was financed by taxes and therefore by the citizens themselves.

2. Transparency

   Of course, availability of Open Government Data increases transparency of governments. It makes decisions comprehensible and may increase public acceptance. Releasing data publicly also demonstrates self-assurance and willingness to handle criticism. Benchmarking is much easier if more data is available.

3. Participation and Collaboration

   Open Government Data increases potential for collaboration of societies. Participation of citizen is immediately increased, open data enables dialog potential between government and citizens. Higher integration of society into the public sector is another result of successful open government initiatives. The study even suggests to reduce consultancy costs by using OGD for crowdsourcing (outsourcing of tasks to communities) of activities.

4. Better Governance

   Open data can lead to better governance. Information can be spread much easier and quicker, feedback can be collected and public discussions can be enabled. Solutions for emerging problems can be found and suggested by participating citizens. Political work can become more efficient. Applications, mashups or services don't have to be developed by the government because people will do this work and might even release their developments and findings on an open source basis.

5. Open Innovation

   Open Government Data provides high potential for innovation. Citizens don't have to ask for the realization of demands, they can just do it themselves and provide their results to others. Unused data could be analyzed and enhanced.

6. Promotion of economic development

   Since companies and industries can use Open Government Data just as everyone else, they can use it to increase their competitiveness and economic output. New products and jobs can be created, new business models can be designed. This also can increase attractiveness as a business location and - last but not least - quality of living for citizens.

3.10 Existing guidelines for Linked Open Government Data

In 2007, 30 Open Government advocates gathered in Sebastopol, California, USA to develop a list of 8 OGD principles [30]. In 2010, the Sunlight Foundation expanded these list to 10 principles [38].

These 10 principles are:

Completeness When publishing open data, nothing should be omitted, all public data should be made available.

Primacy Data should be published as it is collected or stored. It should not be published in aggregated form.

Timeliness To keep the value of the data published, it should be released as early as possible. Information collected should be released as quickly as it is gathered, with priority assigned to data that is time sensitive.

Ease of physical and electronic access Datasets released by the government should be as accessible as possible. No proprietary formats should be used and required user interaction (e.g. filling out forms) should be kept at a minimum. Supplying an API for bulk access is recommended.

Machine readability It is recommended to use formats that can be read by machines. Semantic Web formats are not explicitly mentioned here but since they are designed to be highly machine readable they certainly apply.

Non-discrimination Everyone should be able to access the published data, without requiring registration or identification.

Use of commonly owned standards Data should not be published in formats that require users to buy software to read it. Usage of commonly owned standards increases the number of potential users.

Licensing Maximum openness should be pursued when releasing Open Government Data. Strict licensing terms limit the options of users or developers.

Permanence Information that was published should remain online (this principle was added by the Sunlight Foundation).

Usage Costs There shouldn't be costs for accessing data as this again limits access (principle added by the Sunlight Foundation).

3.11 Examples for Government Data available as LOD

The already discussed LOD cloud also contains data from government agencies. A color-coded version of the diagram is available in which linked government data is easily visible. This data is depicted in figure 3.8 in turquoise color. For example, data from Eurostat, the space agency NASA from the USA and the British government are released as Linked Open Data.

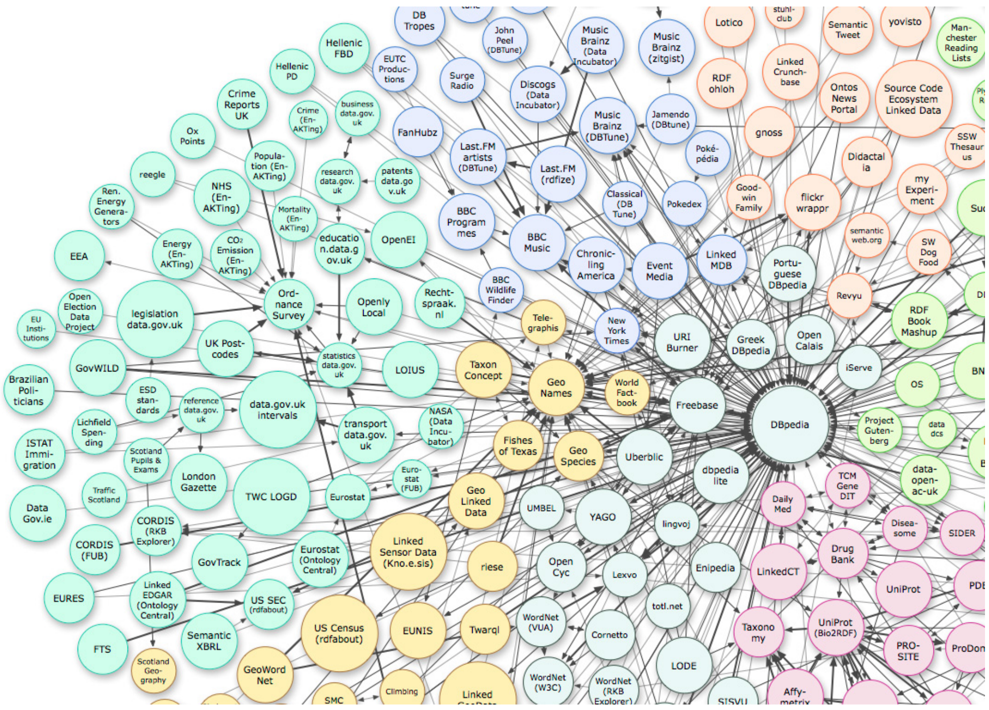


Figure 3.8: Open Government Data within the Linked Open Data cloud (Source: [10])

United States - data.gov

Data.gov is a website that provides government data from the United States of America to the general public. The data released is very comprehensive but is not yet released in a format that conforms to Linked Open Data principles. However, the ”Tetherless World Constellation” at the Rensselaer Polytechnic Institute has created the ”Data-gov Wiki” (actually it is not really a wiki but a website) as an extension that integrates the datasets published at data.gov into the Linked Open Data cloud [13]. They have produced 6.46 billion triples as of early 2013 [57], covering topics like:

- Government spending
- Environmental records
- Statistics on the cost and usage of public services
- Health-related data

The resulting cloud of data is depicted in figure 3.9. The data is available as data files or via SPARQL endpoints. In a scientific article they describe their approach as follows [13]:

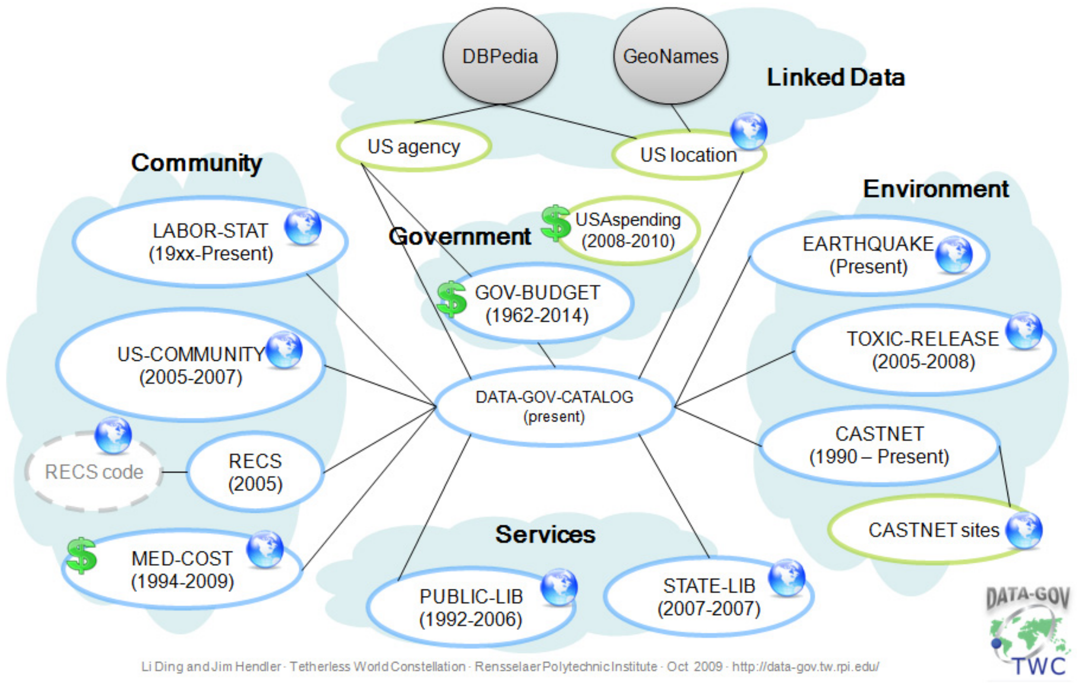


Figure 3.9: LOD-converted Open Government Data from data.gov (Source: [57])

> This [conversion] process is fairly straightforward and highly extensible. First, the raw government data is cleaned-up and preserved through RDF-based representation. Second, these converted datasets use dereferenceable URIs, so that both the datasets and their ontologies can be extended by third party users.
>
> [The enhancement process] focuses on extracting the semantics of literal values in government data into meaningful URIs and linking datasets by declaratively associating URIs mentioned in different datasets. Both automated and manual steps are involved in adding, deriving, linking and integrating government data.

United Kingdom - data.gov.uk

The United Kingdom has already released a variety of Open Government Data. In addition they have also adopted to the standards of Linked Open Data. Contrary to the US portal data.gov they implement and release the LOD implementation by themselves. They have created their own Linked Open Data website where they state: [11].

> Across government over the last ten years there has been a growing realisation to the power of linked data for exposing, sharing, and connecting pieces of data and

information using uniform resource identifiers (URIs).

Since other governments are lacking when it comes to LOD the UK OGD accounts for a substantial share of the government section in the LOD cloud diagram.

Legislation.gov.uk is a good example of how the UK releases Linked Open Data. Legislation.gov.uk is an official government website of the United Kingdom. The website offers access to all published legislation in the UK. The legislation texts can be accessed and shared by users.

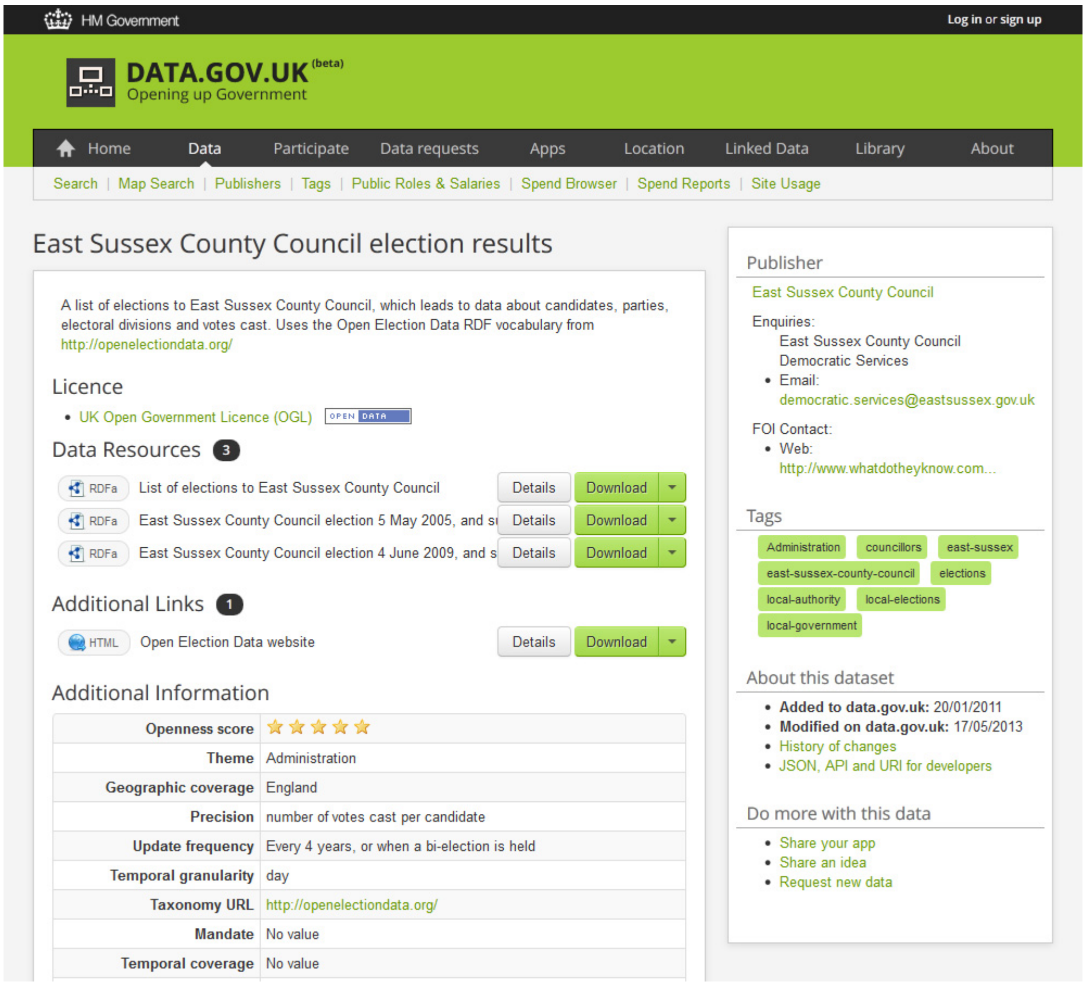
Everything on the portal is available as open data under the terms of an open license (they are using Open Government License [25]) [46]. Data is published with annotations using the RDFa microformat and is also available in RDF. While creating the portal, the UK government followed LOD guidelines like re-using vocabulary where possible and using persistent URIs. W3C standards were used for releasing data. The UK government is using linked data as an underlying technology, the data is also available via an API to developers who want to have programmatic access to data.

Open standards on data.gov.uk

The Government of the United Kingdom officially committed itself to use and re-use open standards and definitions in 2010 [33]. An article published in the same year [46] sees obvious advantages in using linked data instead of conventional databases:

> The most important benefit for publishers of government data is how linked data standards enable departments and agencies to publish their data responsibly. This is because each fact or data point is associated with a URI and that URI can be resolved. The publisher determines what information is returned when a request is made and can serve whatever additional context or provenance information they deem necessary. For example, if the 2002 figures can't be compared with those in subsequent years, the publisher can say so, for every data point. The data can be copied, adapted and re-used, but the publisher always controls what is returned when each URI is dereferenced. This is an important benefit over interchange formats such as CSV or XML where data can be changed or context lost as it is passed from hand to hand or system to system.

The authors come to the conclusion that following strict standards is an advantage when gathering and generation OGD. Other advantages mentioned aim toward flexibility. Linked data can be used and published in a very modular or atomic way. It can evolve and not all details


HM Government

Log in or sign up

DATA.GOV.UK (beta)
Opening up Government

Home | Data | Participate | Data requests | Apps | Location | Linked Data | Library | About

Search | Map Search | Publishers | Tags | Public Roles & Salaries | Spend Browser | Spend Reports | Site Usage

East Sussex County Council election results

A list of elections to East Sussex County Council, which leads to data about candidates, parties, electoral divisions and votes cast. Uses the Open Election Data RDF vocabulary from http://openelectiondata.org/

Licence

- UK Open Government Licence (OGL) OPEN DATA

Data Resources 3

- RDFa List of elections to East Sussex County Council — Details — Download
- RDFa East Sussex County Council election 5 May 2005, and s — Details — Download
- RDFa East Sussex County Council election 4 June 2009, and s — Details — Download

Additional Links 1

- HTML Open Election Data website — Details — Download

Additional Information

Openness score	★★★★★
Theme	Administration
Geographic coverage	England
Precision	number of votes cast per candidate
Update frequency	Every 4 years, or when a bi-election is held
Temporal granularity	day
Taxonomy URL	http://openelectiondata.org/
Mandate	No value
Temporal coverage	No value

Publisher

East Sussex County Council

Enquiries:
East Sussex County Council
Democratic Services
- Email: democratic.services@eastsussex.gov.uk

FOI Contact:
- Web: http://www.whatdotheyknow.com...

Tags

Administration, councillors, east-sussex, east-sussex-county-council, elections, local-authority, local-elections, local-government

About this dataset

- **Added to data.gov.uk:** 20/01/2011
- **Modified on data.gov.uk:** 17/05/2013
- History of changes
- JSON, API and URI for developers

Do more with this data

- Share your app
- Share an idea
- Request new data

Figure 3.10: Data.gov.uk features data in RDFa format as well as rating of how well a dataset complies to LOD guidelines

have to be planned in advance. Also, portability of linked data is mentioned. By deciding to use linked data as a foundation for their service, the UK administration is not locked in on any proprietary platforms or technologies.

Figure 3.10 shows how data is published by the UK portal on the example of an election result. A publicly available vocabulary (Open Election Data RDF) is used. A license is stated and the data is available in RDFa format. The website even features an ”Openness score” where they use the star schema developed by Tim Berners-Lee and described earlier in this chapter to rate data between 1 and 5 stars. The dataset shown in the figure qualifies as 5-star data. The website also defines a persistent URI (not shown in the screenshot) and allows to download data in other formats or via an API for developers.

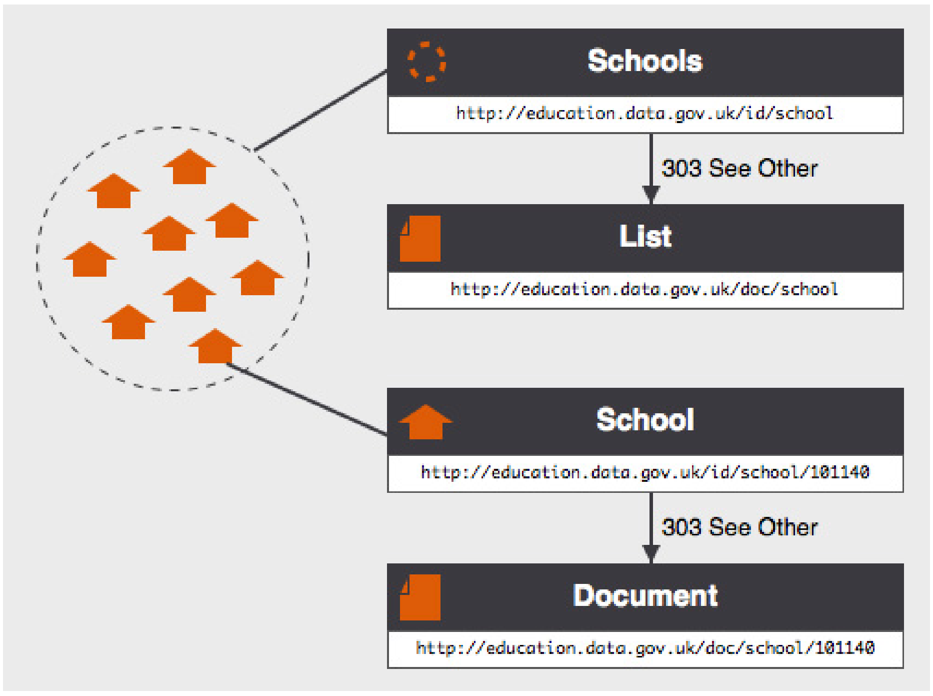

Schools
http://education.data.gov.uk/id/school
303 See Other
List
http://education.data.gov.uk/doc/school
School
http://education.data.gov.uk/id/school/101140
303 See Other
Document
http://education.data.gov.uk/doc/school/101140

Figure 3.11: The principle for URI design at data.gov.uk (Source: [52])

3.12 Implementation of data.gov.uk

URI design was very carefully planned for data.gov.uk. As shown in figure 3.11 there are 4 kinds of resources that are considered when it comes to real-world-things: URIs for the real-world things themselves, URIs for the documents about those real-world things, URIs for sets of real-world things of the same type, and URIs for documents about those sets [52]. The schemata for building the corresponding URIs are:

Real-world things (e.g. a school):

```
http://{sector}.data.gov.uk/id/{concept}[/{identifier}]
```

Documents about those things (there is a 303 HTTP forward from a thing to the corresponding document):

```
http://{sector}.data.gov.uk/doc/{concept}[/{identifier}]
```

Vocabularies, classes, properties, concept schemes and concepts:

```
http://{sector}.data.gov.uk/def/{scheme}[/{concept}]
```

Datasets and the graphs they contain:

`http://{sector}.data.gov.uk/data/{dataset}[/{part}] for`

A problem that occurred when creating the schemata and datasets was versioning [46]. If a name of a thing occurs in several databases that have different update intervals inconsistency might result from a name change between such intervals. Named graphs were adopted to solve this problem. The name of a thing will only be stored once.

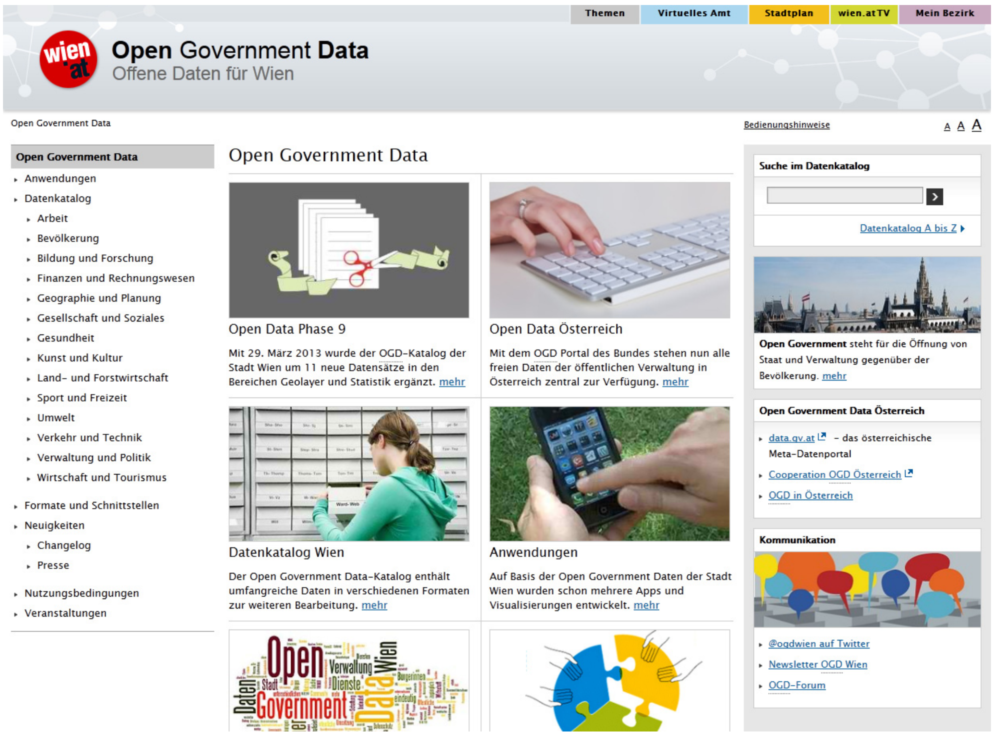
The UK government has also supported the creation of middleware to provide data views based on SPARQL data. This includes XML and JSON views as well as API creation. As a result all UK government linked data is also available through RESTful APIs.

3.13 Open Government Data in Vienna

The focus of this thesis is Open Government Data in Vienna. This data is available from a web platform [31]. Figure 3.12 shows a screenshot of this web portal. The city government of Vienna started publishing open data in mid-2011. They already operated an extensive online city map service at this time, including many ”Points Of Interest” (POI). A POI can be any place or location, like (for example) a school, a museum or a park. Therefore, a first release of open data could be realized within months. The OGD website [32] states that there are 127 information layers available for their city map.

Most datasets are available in various formats. The primary format is CSV and RSS/XML, however there are other formats available such as WMS, WFS or JSON. The OGD website also states that a Linked Open Data release is currently under evaluation. This was confirmed with the Vienna OGD team, however, as of early 2013, there was no schedule for a release. Of course, it is likely that a complete and maintained release will follow at some point and that the City Government of Vienna or its contractors will face similar challenges to the ones that are dealt with in the next chapter of this thesis.

The City Government of Vienna seems to handle the issue of open data in a very professional way. There are regular extensions and changes to the dataset. Every few months a new release cycle is initiated, usually containing new datasets and optimizations to the schemata of existing ones. There is a change log available online that lists all changes back to the very first release. The city government of Vienna also takes a lot of effort to integrate the community of users and developers into the project. A meeting with this community is organized every three months where new developments are presented and discussed. Everyone interested can participate in that meetings and video recordings are put online afterwards. In addition to that, the OGD team


Themen | Virtuelles Amt | Stadtplan | wien.at TV | Mein Bezirk

Open Government Data
Offene Daten für Wien

Open Government Data

Bedienungshinweise A A A

Open Government Data
- Anwendungen
- Datenkatalog
  - Arbeit
  - Bevölkerung
  - Bildung und Forschung
  - Finanzen und Rechnungswesen
  - Geographie und Planung
  - Gesellschaft und Soziales
  - Gesundheit
  - Kunst und Kultur
  - Land- und Forstwirtschaft
  - Sport und Freizeit
  - Umwelt
  - Verkehr und Technik
  - Verwaltung und Politik
  - Wirtschaft und Tourismus
- Formate und Schnittstellen
- Neuigkeiten
  - Changelog
  - Presse
- Nutzungsbedingungen
- Veranstaltungen

Open Government Data

Open Data Phase 9

Mit 29. März 2013 wurde der OGD-Katalog der Stadt Wien um 11 neue Datensätze in den Bereichen Geolayer und Statistik ergänzt. mehr

Open Data Österreich

Mit dem OGD Portal des Bundes stehen nun alle freien Daten der öffentlichen Verwaltung in Österreich zentral zur Verfügung. mehr

Datenkatalog Wien

Der Open Government Data-Katalog enthält umfangreiche Daten in verschiedenen Formaten zur weiteren Bearbeitung. mehr

Anwendungen

Auf Basis der Open Government Daten der Stadt Wien wurden schon mehrere Apps und Visualisierungen entwickelt. mehr

Suche im Datenkatalog

Datenkatalog A bis Z

Open Government steht für die Öffnung von Staat und Verwaltung gegenüber der Bevölkerung. mehr

Open Government Data Österreich
- data.gv.at – das österreichische Meta-Datenportal
- Cooperation OGD Österreich
- OGD in Österreich

Kommunikation
- @ogdwien auf Twitter
- Newsletter OGD Wien
- OGD-Forum

Figure 3.12: The Vienna Open Goverment Data web portal in May 2013

operates a public forum on their website and they use a twitter channel to communicate.

These community efforts seem to be successful since there are more than 80 applications available as of May 2013. These applications are prominently listed on the Vienna OGD website. Most of them are visualizations or Apps for mobile operating systems like Apples iOS or Googles Android platform. Many applications are freely available, however some of them need to be purchased. It is notable that most applications use only a very small subset of the data available. At least 2 of these applications are based on Semantic Web technologies. However, these applications are prototypical implementations as there is no extensive Linked Open Data implementation yet.

Vienna OGD also integrates data from other sources. Data providers can submit data that will be added to the portal if certain criteria are met. However this integration is not done for free.

Available data at Vienna OGD

All open data from the Vienna OGD platform is released under the ”Creative Commons Attribution 3.0” license, the least restrictive Creative Commons license available. According to the license terms [9], users are free to share and adapt the open data in a commercial or non-commercial way. The only requirement is that users attribute the work and refer to the city government of Vienna as data source.

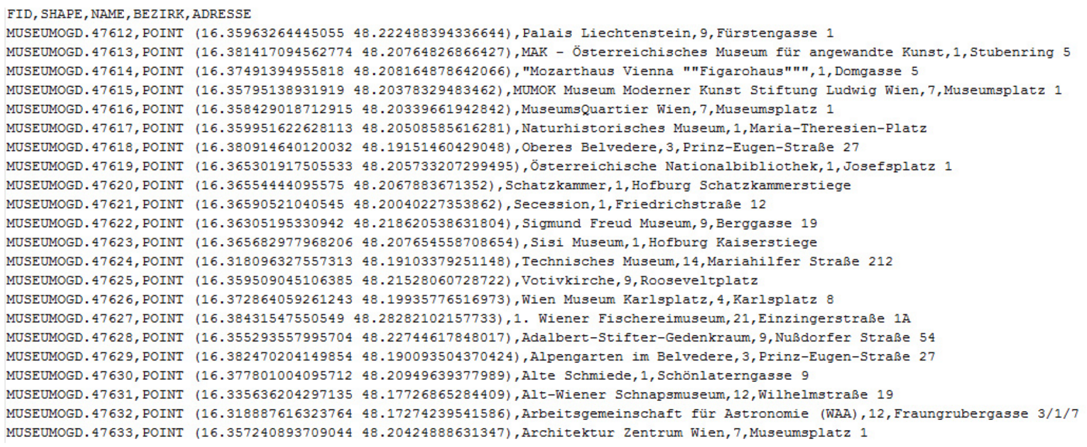
Since German is the official language in Austria all the data available at Vienna OGD is only available in German language. This applies to the website as well as the datasets made available. No data is released in English language and no translations are provided. Translation of the data into other language will not be part of this thesis. Since there are many specific words and geographic names, an automated translation of such data doesn't seem to be an option when data quality is taken into account.

The open data at the Vienna OGD platform can roughly be divided into two distinct types. The first type is statistical and numerical data. Examples are:

- Extensive demographic data
- Budget data
- Data about the employment market
- Tax statistics
- Names that are given to newborn children
- Automobile registrations

The second type are Points of Interest divided into several layers. Examples are:

- Schools
- Hospitals
- Parks
- Playgrounds
- Sights
- Museums



```
FID,SHAPE,NAME,BEZIRK,ADRESSE
MUSEUMOGD.47612,POINT (16.35963264445055 48.222488394336644),Palais Liechtenstein,9,Fürstengasse 1
MUSEUMOGD.47613,POINT (16.381417094562774 48.20764826866427),MAK - Österreichisches Museum für angewandte Kunst,1,Stubenring 5
MUSEUMOGD.47614,POINT (16.37491394955818 48.208164878642066),"Mozarthaus Vienna ""Figarohaus""",1,Domgasse 5
MUSEUMOGD.47615,POINT (16.35795138931919 48.20378329483462),MUMOK Museum Moderner Kunst Stiftung Ludwig Wien,7,Museumsplatz 1
MUSEUMOGD.47616,POINT (16.358429018712915 48.20339661942842),MuseumsQuartier Wien,7,Museumsplatz 1
MUSEUMOGD.47617,POINT (16.359951622628113 48.20508585616281),Naturhistorisches Museum,1,Maria-Theresien-Platz
MUSEUMOGD.47618,POINT (16.380914640120032 48.19151460429048),Oberes Belvedere,3,Prinz-Eugen-Straße 27
MUSEUMOGD.47619,POINT (16.365301917505533 48.205733207299495),Österreichische Nationalbibliothek,1,Josefsplatz 1
MUSEUMOGD.47620,POINT (16.36554444095575 48.2067883671352),Schatzkammer,1,Hofburg Schatzkammerstiege
MUSEUMOGD.47621,POINT (16.36590521040545 48.20040227353862),Secession,1,Friedrichstraße 12
MUSEUMOGD.47622,POINT (16.36305195330942 48.218620538631804),Sigmund Freud Museum,9,Berggasse 19
MUSEUMOGD.47623,POINT (16.365682977968206 48.207654558708654),Sisi Museum,1,Hofburg Kaiserstiege
MUSEUMOGD.47624,POINT (16.318096327557313 48.19103379251148),Technisches Museum,14,Mariahilfer Straße 212
MUSEUMOGD.47625,POINT (16.359509045106385 48.21528060728722),Votivkirche,9,Rooseveltplatz
MUSEUMOGD.47626,POINT (16.372864059261243 48.19935776516973),Wien Museum Karlsplatz,4,Karlsplatz 8
MUSEUMOGD.47627,POINT (16.38431547550549 48.28282102157733),1. Wiener Fischereimuseum,21,Einzingerstraße 1A
MUSEUMOGD.47628,POINT (16.355293557995704 48.22744617848017),Adalbert-Stifter-Gedenkraum,9,Nußdorfer Straße 54
MUSEUMOGD.47629,POINT (16.382470204149854 48.190093504370424),Alpengarten im Belvedere,3,Prinz-Eugen-Straße 27
MUSEUMOGD.47630,POINT (16.377801004095712 48.20949639377989),Alte Schmiede,1,Schönlaterngasse 9
MUSEUMOGD.47631,POINT (16.335636204297135 48.17726865284409),Alt-Wiener Schnapsmuseum,12,Wilhelmstraße 19
MUSEUMOGD.47632,POINT (16.318887616323764 48.17274239541586),Arbeitsgemeinschaft für Astronomie (WAA),12,Fraungrubergasse 3/1/7
MUSEUMOGD.47633,POINT (16.357240893709044 48.20424888631347),Architektur Zentrum Wien,7,Museumsplatz 1
```

Figure 3.13: CSV data from the Vienna OGD portal

Available formats and data at Vienna OGD

Statistical and numerical data is available in CSV (Comma Separated Values) format only.

POI datasets are available in a variety of formats. Available formats are:

GML Geography Markup Language - an XML vocabulary defined by the Open Geospatial Consortium to express geographical places

JSON JavaScript Object Notation - a text-based standard to describe data in a ways that it stays readable for humans (in contrast to RDF, which is hard to read for humans)

KML Keyhole Markup Language - an XML-based notation for geographic places, widely used in map applications like Google Earth

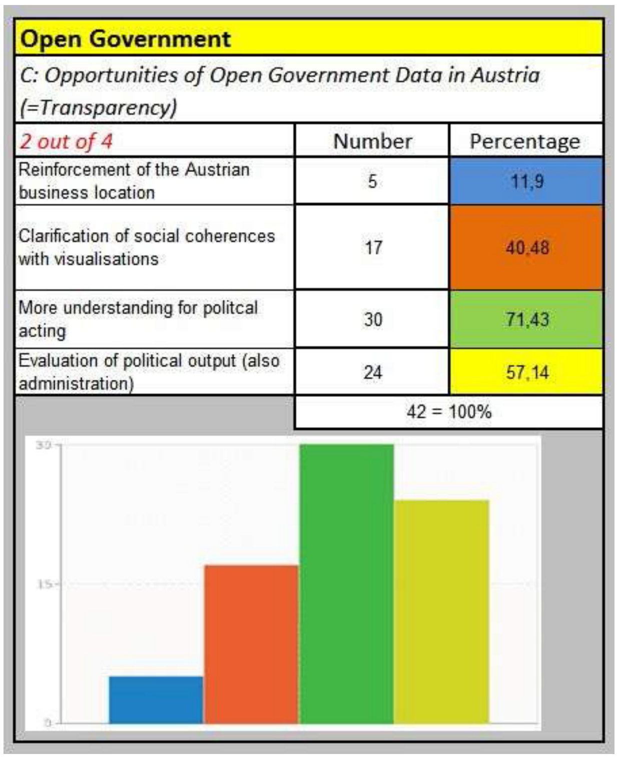
GeoRSS GeoRSS is a standard for integrating geographical places into RSS feeds

CSV Comma Separated Values - a very efficient data format based on a spreadsheet layout, with data divided into lines and columns

ESRI Shapefile A data format used to describe geographical places, widely used in geographic information system (GIS) software

These data formats are clearly targeted toward geographic data. The only data format available for all datasets of both types is CSV. An excerpt of a CSV dataset is depicted in figure 3.13 (museums in Vienna).

Vienna OGD - Evaluation



Open Government

C: Opportunities of Open Government Data in Austria (=Transparency)

2 out of 4	Number	Percentage
Reinforcement of the Austrian business location	5	11,9
Clarification of social coherences with visualisations	17	40,48
More understanding for politcal acting	30	71,43
Evaluation of political output (also administration)	24	57,14
	42 = 100%	

Figure 3.14: How OGD is seen in Austria (Source: [17])

In general, the Vienna OGD initiative offers an extensive list of datasets. The data is updated on a regular basis and grows quickly. Efforts to communicate the availability and advantages of open data were successful so far. The high (and growing) number of applications that were already created by the community are an indicator for that.

Regarding data and formats, it is noticeable that many datasets are offered in a wide variety of formats. They all are views on the same source data however. Where the Vienna OGD initiative is currently lacking is uniformity of data and the missing adoption of LOD standards. Both issues are connected. It is observable from the data.gov.uk project that using LOD standards increases uniformity and therefore usefulness of the data.

This thesis will create a Linked Open Data implementation by converting the available data. This is a comparable approach to what was done by the data-gov Wiki in the USA. The more sustainable approach of course would be to use linked data standards and guidelines right from the start. This however would take a much higher effort and would basically be a re-implementation of the Vienna OGD initiative.

A study published in 2011 evaluates Open Government Data in Austria [17]. The authors conducted workshops with participants from politics, industry, administration and civil society. Some results are depicted in figure 3.14. It is visible that transparency and evaluation of political output are seen as more important opportunities than economic advantages.

Vienna OGD - Evaluation of data quality

Before the Linked Open Data implementation was started, an extensive evaluation of the data quality was conducted. During this evaluation, several issues emerged. In general the data quality is high when only isolated datasets are taken into account. This is because diversity and availability of datasets are high and data is available in a variety of formats.

When it comes to interoperability, semantics or compatibility of the data, the uniformity is much lower.

For several datasets that contain geographical places (POIs), the attribute assignments were evaluated. Figure 3.15 shows the resulting matrix. Only two attributes are available in every dataset. These two attributes are the ID of a place and its geographic location (geodata in form of latitude/longitude). District number, address and weblink occur often, but not in all datasets. All other attributes are only used in some isolated datasets.

When it comes to attributes used in the datasets, it is notable that different attributes are semantically at least similar. For example, several attributes like ”name” or ”bezeichnung” are used to label objects. There are also several attributes that are used to describe the type or category of an object, e.g. ”typ”, ”kategorie”, ”kategorie_txt” or ”mk_txt”. A reason for this diversity might be that the data is taken from different sources or even different agencies or offices where different attributes and labels are used. There are even some datasets where no label at all is used and only ID and geographic position are included in the dataset. There is also a variety of datasets with several attributes that just hold some further descriptions and information about an object.

	Büchereien	Kindergärten	Multimediastationen	Musik- und Singschulen	Schulen	Universitäten und Fachhochschulen	Volkshochschulen	Ambulanzen	Defibrillatoren	Fundboxen	Krankenhäuser	Sozialmärkte	Sportstätten	Wohn- und Pflegehäuser	Burgen und Schlösser	Kunstwerke im Öff. Raum	Museen und Sammlungen	Parkanlagen	Badestellen	Campingplätze	Fahradabstellanlagen	Grillplätze	Hundezonen	Schwimmbäder	Spielplätze	Carsharing-Standorte	Fahrschulen	Sehenswürdigkeiten
FID	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SHAPE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
NAME	X			X	X	X	X								X		X							X			X	X
BEZEICHNUNG		X				X		X			X	X		X					X							X		
KRANKENHAUS_BEZEICHNUNG								X																				
BEZIRK	X		X	X			X	X	X	X	X	X		X			X	X	X	X	X			X	X	X	X	X
ADRESSE				X			X			X		X					X				X					X	X	
PLZ																X												
ORT																X										X		
STRASSE																X				X								
HAUSNUMMER																				X								
TELEFON	X			X			X													X								
E-MAIL	X			X			X																					
INTERNET																												X
WEBLINK	X	X		X			X	X		X		X	X	X						X		X	X	X	X	X	X	
INFO									X																			
ZUSATZ			X																									
STANDORT_FUNDBOX										X																		
STANDORT																X									X			
LAGE																						X						
STOCK									X																			
SICHTBAR															X													
VERLAUF																												
OEFFNUNGSZEITEN 1-6	X			X			X											X							X			
HINWEIS									X																			
TYP		X														X							X					
KATEGORIE_TXT													X															
KATEGORIE															X											X		
MK_TXT																											X	
SPORTSTAETTEN_ART													X															
FLAECHE													X					X					X					
LANGTEXT											X	X		X														
WEITERE_INFO																			X									
BESCHREIBUNG															X													
INFORMATION																												
STELLPL_ANZ																												
FZG_ANZAHL																										X		
ANZAHL																					X							
STATION																												
BETREIBER		X																										
ABTEILUNG								X																				
BEZIRKSAMT										X																		
PARKANLAGE																		X					X					
SPIELPLATZ																		X										
BETRIEB																				X								
RESERVIERUNG																						X						
EINFRIEDUNG																							X					
MERKMAL																												
WASSER																									X			
ZEITRAUM																												

Figure 3.15: Attributes assigned to Vienna OGD datasets

Another issue is the address data. Many objects contain contact and address data in some form. However, there is no homogeneous way of storing this kind of data. For some objects, address data is stored in a normalized form, where the data is distributed among several attributes containing for example the post code or the street address:

```
PLZ: 1120
ORT: "Wien"
STRASSE: "Vor Meidlinger Hauptstrasse 3"
STANDORT: "In Gruenflaeche zwischen zwei Baeumen"
```

However, in some cases the post code is not available, instead the district number (that could be used to generate the post code) is stored along with the street address:

```
BEZIRK: 2
ADRESSE: "Ernst-Melchior-Gasse 9"
```

And in some cases this information is condensed into a single attribute:

```
ADRESSE: "10., Ettenreichgasse 45B"
```

A direct conversion of this data would be problematic because available vocabulary for storing address data demands for higher data quality.

In addition, the datasets are not interlinked, although they could be to some extent. There are no stable connections between individual datasets. An example where these connections would make sense are playgrounds. In Vienna, many playgrounds are located within a park. The playground dataset contains an attribute ”standort” that lists the label of parks, but there is no stable link using e.g. the IDs.

Finally, there are no links or references to other open data sources at all. Of course, since the Vienna OGD is not designed to be Linked Open Data, this was to be expected.

3.14 Research question

The question that will be addressed in the next chapter is: How can a set of heterogeneous Open Government Data be integrated into the cloud of Linked Open Data on the web while maintaining LOD requirements and re-using vocabulary without redesigning the source dataset?

To summarize, weaknesses that need to be taken care of during a conversion are:

- Usage of distinct attributes that are semantically similar
- Address data is stored in various ways
- No direct connections between objects

- No links to other data sources

The main issue however is that the datasets are currently isolated and do not follow a common schema. All of these issues will be addressed in the conversion process.

CHAPTER 4

Solution

4.1 Result requirements

The goal of the thesis work is to convert a subset of the Vienna OGD initiative into usable LOD. According to Linked Open Data guidelines [19] the following requirements should be fulfilled for the data to qualify as 5-star ”Linked Open Data” in the first place:

1. The data has to be available on the web with an open license
2. It has to be available as machine-readable structured data in a non-proprietary format
3. Open standards from the W3C have to used
4. The data has to be linked to other data sources

In addition to that official requirements the following requirements were defined to ensure that the resulting data will be useful:

5. The resulting dataset should consist of at least 100,000 triples including metadata
6. The aforementioned uniformity issues should be resolved
7. There should be internal links (object properties between resources) in addition to the external links
8. The dataset should follow a common model in form of an OWL ontology
9. Existing vocabulary should be used for properties if possible

10. The data should be published on a server so it can be queried using SPARQL

After the conversion process, the results will be evaluated against this requirements.

The W3C published a guideline document [8] for creating Linked Open Data. The conversion process loosely follows this document.

4.2 Selecting data categories

The Vienna OGD initiative offers a variety of available datasets. There is financial and demographic data available as well as statistical data (cars, names for newborns, etc.) and data about POIs (Points of Interest). The last section is the most comprehensive, therefore the transformation into LOD will focus on these datasets. Transferring all the data into Linked Open Data would be a very extensive project due to the varying forms of content and data, schema generation and maintenance would be a very complex task that exceeds the focus of a thesis.

The datasets are available in various formats like CSV, RSS/XML or JSON. For downloading the source data, the CSV format is used for performance reasons (smaller file sizes and download times) and because it is the only format that all datasets are published in.

Regarding the POI data, a subset of 29 categories was selected. Included within this subset are all categories that feature geographical point data. This excludes all datasets that include path or surface data like hiking trails or bicycle routes. Some categories were dropped because their CSV export contained errors (for examples line breaks within description texts). The 29 categories that were selected for the transformation are:

Type	Identifier	Type (German)
Camping sites	CAMPINGPLATZOGD	Campingplatz
Parks	PARKANLAGEOGD	Parkanlage
Dog zones	HUNDEZONEOGD	Hundezone
Playgrounds	SPIELPLATZOGD	Spielplatz
Barbecue areas	GRILLPLATZOGD	Grillplatz
Public swimming pools	SCHWIMMBADOGD	Schwimmbad
Museums	MUSEUMOGD	Museum
Universities	UNIVERSITAETOGD	Universität
Carsharing spaces	CARSHARINGOGD	Carsharing-Standort
Driving schools	FAHRSCHULEOGD	Fahrschule
Sights	SEHENSWUERDIGOGD	Sehenswürdigkeit
Libraries	BUECHEREIOGD	Bücherei

Kindergartens	KINDERGARTENOGD	Kindergarten
Multimedia stations	MULTIMEDIAOGD	Multimedia-Station
Music schools	MUSIKSINGSCHULEOGD	Musikschule
Schools	SCHULEOGD	Schule
Adult education centers	VOLKSHOCHSCHULEOGD	Volkshochschule
Walk-in clinics	AMBULANZOGD	Ambulanz
Defibrillators	DEFIBRILLATOROGD	Defibrillator
Lost&Found boxes	FUNDBOXOGD	Fundbox
Hospitals	KRANKENHAUSOGD	Krankenhaus
Shops for people with low income	SOZIALMARKTOGD	Sozialmarkt
Sports centers	SPORTSTAETTENOGD	Sportstätte
Nursing homes	WOHNPFLEGEHAUSOGD	Pflegehaus
Castles	BURGSCHLOSSOGD	Burg/Schloss
Bathing areas	BADESTELLENOGD	Badestelle
Bicycle racks	FAHRRADABSTELLANLAGEOGD	Fahradabstellanlage
Police stations	POLIZEIOGD	Polizeistation
Cemeteries	FRIEDHOFOGD	Friedhof

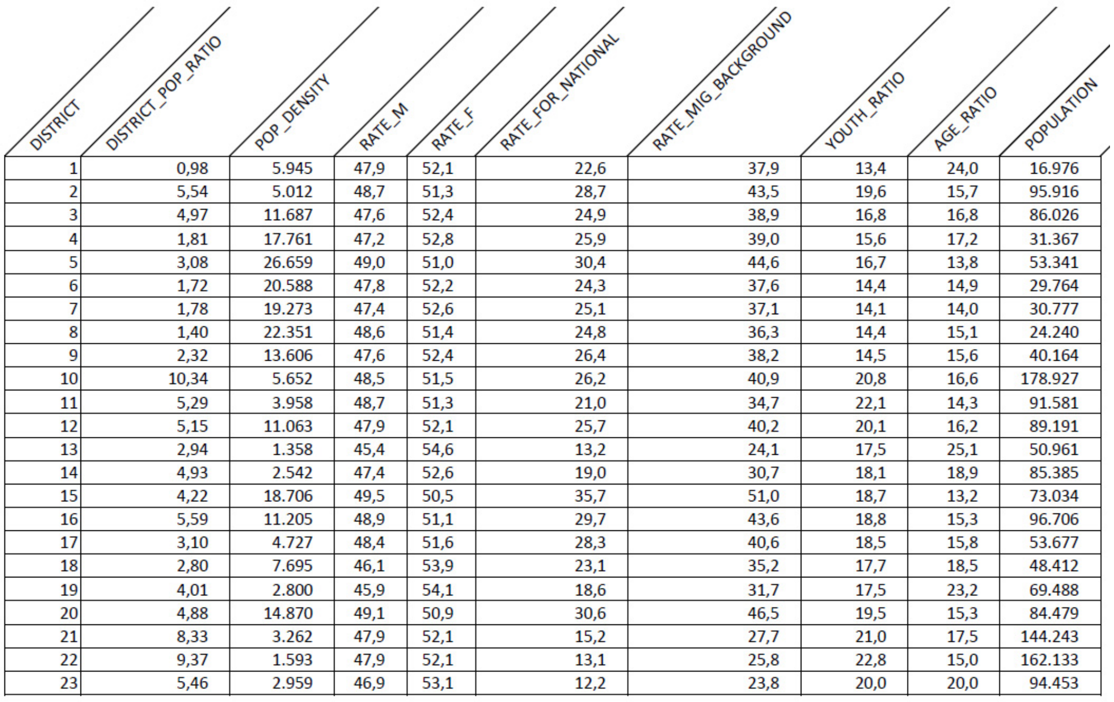
The URL of the CSV files that were used can be generated and derived from the identifier by using the following schema:

```
http://data.wien.gv.at/daten/geoserver/ows?service=WFS&amp;request=
GetFeature&amp;version=1.1.0&amp;typeName=ogdwien:IDENTIFIER&
amp;srsName=EPSG:4326&amp;outputFormat=csv
```

Where ”IDENTIFIER” needs to be replaced by the type (e.g. ”MUSEUMOGD”).

To demonstrate the addition of schematically diverging data, some demographic data was also included within the transformation. The Vienna OGD platform contains a dataset that lists this demographic data for every district. This dataset is only available in CSV format. The following demographic indicators are added to the resulting Linked Open Data:

- Population ratio DISTRICT_POP_RATIO)
- Population density (POP_DENSITY)
- Rate of male population (RATE_M)
- Rate of female population (RATE_F)



DISTRICT	DISTRICT_POP_RATIO	POP_DENSITY	RATE_M	RATE_F	RATE_FOR_NATIONAL	RATE_MIG_BACKGROUND	YOUTH_RATIO	AGE_RATIO	POPULATION
1	0,98	5.945	47,9	52,1	22,6	37,9	13,4	24,0	16.976
2	5,54	5.012	48,7	51,3	28,7	43,5	19,6	15,7	95.916
3	4,97	11.687	47,6	52,4	24,9	38,9	16,8	16,8	86.026
4	1,81	17.761	47,2	52,8	25,9	39,0	15,6	17,2	31.367
5	3,08	26.659	49,0	51,0	30,4	44,6	16,7	13,8	53.341
6	1,72	20.588	47,8	52,2	24,3	37,6	14,4	14,9	29.764
7	1,78	19.273	47,4	52,6	25,1	37,1	14,1	14,0	30.777
8	1,40	22.351	48,6	51,4	24,8	36,3	14,4	15,1	24.240
9	2,32	13.606	47,6	52,4	26,4	38,2	14,5	15,6	40.164
10	10,34	5.652	48,5	51,5	26,2	40,9	20,8	16,6	178.927
11	5,29	3.958	48,7	51,3	21,0	34,7	22,1	14,3	91.581
12	5,15	11.063	47,9	52,1	25,7	40,2	20,1	16,2	89.191
13	2,94	1.358	45,4	54,6	13,2	24,1	17,5	25,1	50.961
14	4,93	2.542	47,4	52,6	19,0	30,7	18,1	18,9	85.385
15	4,22	18.706	49,5	50,5	35,7	51,0	18,7	13,2	73.034
16	5,59	11.205	48,9	51,1	29,7	43,6	18,8	15,3	96.706
17	3,10	4.727	48,4	51,6	28,3	40,6	18,5	15,8	53.677
18	2,80	7.695	46,1	53,9	23,1	35,2	17,7	18,5	48.412
19	4,01	2.800	45,9	54,1	18,6	31,7	17,5	23,2	69.488
20	4,88	14.870	49,1	50,9	30,6	46,5	19,5	15,3	84.479
21	8,33	3.262	47,9	52,1	15,2	27,7	21,0	17,5	144.243
22	9,37	1.593	47,9	52,1	13,1	25,8	22,8	15,0	162.133
23	5,46	2.959	46,9	53,1	12,2	23,8	20,0	20,0	94.453

Figure 4.1: The selection of demographic data from the Vienna OGD platform

- Rate of people with foreign nationality (RATE_FOR_NATIONAL)
- Rate of people with migration background (RATE_MIG_BACKGROUND)
- Rate of young people (age <= 19; YOUTH_RATIO)
- Rate of elderly people (age >= 65; AGE_RATIO)
- Total population (POPULATION)

A list of the dataset from the year 2012 containing demographic data for all 23 districts in Vienna is represented in figure 4.1. This data will be added to the resulting dataset on a per-district basis.

4.3 Introducing a common model

Now that the source data that will be utilized was selected, a common model for the resulting RDF data can be created. This kind of model comes in form of an OWL ontology. Protege was

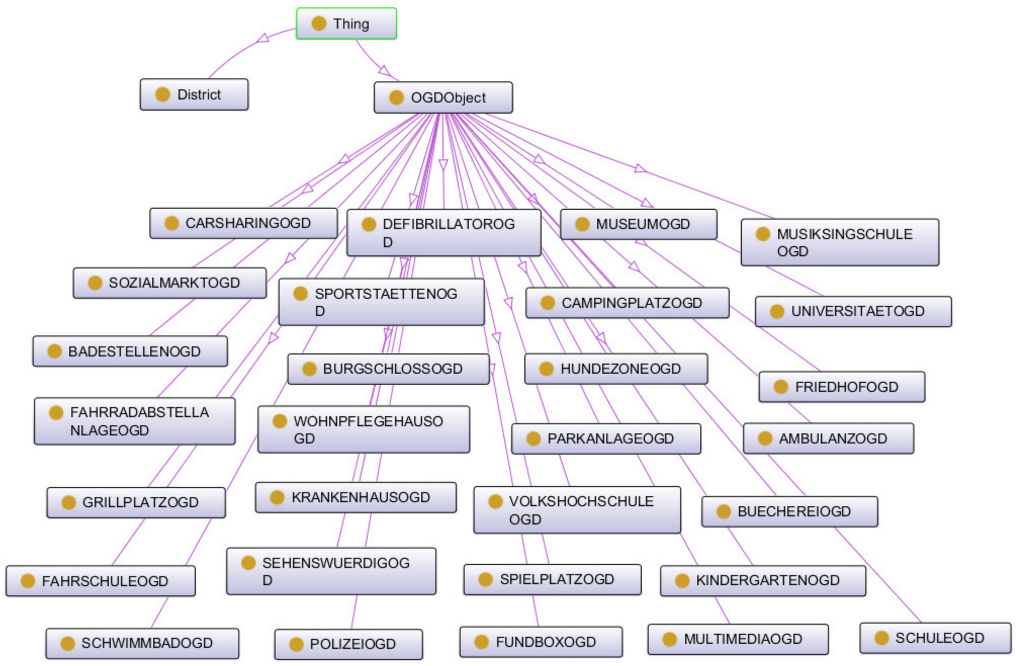


Figure 4.2: Vienna OGD class hierarchy

used to design the ontology. There are 2 main classes. The class "OGDObject" is designed as a base class for POI objects. Most individuals therefore will have OGDObject as a superclass. However, there will be a distinct class for every type of POI that is included. These distinct classes are defined as subclasses of OGDObject. There is a subclass for every type of POI. The names of these subclasses are identical to the identifiers that are used within the OGD Vienna source data (for example "MUSEUMOGD" for museums). Since every POI has to be within a certain district of Vienna and this data is available for many objects there will be another class representing a district. This class is named "District". It does not have subclasses or superclasses (except for owl:Thing of course). Since there are 23 districts within Vienna the resulting RDF graph will contain 23 individuals of this class.

The ontology classes and their hierarchy is depicted in figure 4.2.

A summary of the classes defined in the ontology:

owl:Thing Superclass of all user-defined classes in OWL.

District Represents a district.

OGDObject Superclass of all objects (POI) from OGD Vienna.

Subclasses of OGDObject The subclasses of OGDObject represent the defined types of POIs. There are 29 classes named CAMPINGPLATZOGD, PARKANLAGEOGD, HUNDEZONEOGD, SPIELPLATZOGD, GRILLPLATZOGD, SCHWIMMBADOGD, MUSEUMOGD, UNIVERSITAETOGD, CARSHARINGOGD, FAHRSCHULEOGD, SEHENSWUERDIGOGD, BUECHEREIOGD, KINDERGARTENOGD, MULTIMEDIAOGD, MUSIKSINGSCHULEOGD, SCHULEOGD, VOLKSHOCHSCHULEOGD, AMBULANZOGD, DEFIBRILLATOROGD, FUNDBOXOGD, KRANKENHAUSOGD, SOZIALMARKTOGD, SPORTSTAETTENOGD, WOHNPFLEGEHAUSOGD, BURGSCHLOSSOGD, BADESTELLENOGD, FAHRRADABSTELLANLAGEOGD, POLIZEIOGD and FRIEDHOFOGD.

4.4 URIs

Unfortunately, a perfect solution for defining a URI namespace for the use case is not possible. Such a solution would be to have URIs that point to the Vienna OGD server (data.wien.gv.at). However, since we don't have control over this server it wouldn't make too much sense to define such URIs. The W3C LOD Cookbook [8] explicitly states:

> It is important to select a DNS domain that your department or agency controls. It is bad etiquette to use someone's domain that you do not own or control. In this way, you can also commit to its permanence and provide data at this address.

Therefore the namespace for URIs was defined to be

```
http://ogd.ifs.tuwien.ac.at/vienna
```

The disadvantage of this namespace is that it points to a university server and therefore cannot be identified as containing Open Government Data at first sight. However, the big advantage is that this server can be controlled by university staff and therefore using it as a namespace is much more future proof for maintaining the implementation.

4.5 Using existing schemas

The definition of properties is more complex. The initially defined requirement that states that existing vocabulary should be re-used if possible is especially important for the properties (predicates).

Labeling the individuals

District and OGDObject as well as its subclasses use the rdfs:label property to label the individuals. An alternative would be foaf:name (FOAF vocabulary) but the rdfs:label property is more commonly used for labeling individuals, e.g. by DBpedia.

Property	label
Type	Data Property
Namespace	http://www.w3.org/2000/01/rdf-schema#
Description	The human readable label of an individual.
Domain	OGDObject, District
Data type	String

Properties defining the OGDObject individuals

All POIs within Vienna OGD have an ID attached. This ID is of the form TYPE.X, where TYPE denotes the OGD type (i.e. MUSEUMOGD) and X is a progressing number attached to the individual object. The decision was taken to use this ID in unmodified form within the Linked Open Data. The reason for doing so is compatibility, e.g. when data from other sources based on Vienna OGD should be linked with the results from this thesis. The Dublin Core vocabulary contains the property ”identifier” that is re-used for this purpose.

Property	identifier
Type	Data Property
Namespace	http://purl.org/dc/elements/1.1/
Description	A unique identifier for every POI object
Domain	OGDObject
Data type	string

When it comes to POIs, the geographic location is extremely important. All POIs from the Vienna OGD have their exact coordinates attached. The WGS84 ontology that defines terms for latitude (lat) and longitude (long) uses WGS84 as a reference datum and is very well suited for this use case. This vocabulary is also widely used (e.g. by DBpedia).

Property	lat
Type	Data Property
Namespace	http://www.w3.org/2003/01/geo/wgs84_pos#
Description	Geographic latitude of an object
Domain	OGDObject
Data type	string

Property	long
Type	Data Property
Namespace	http://www.w3.org/2003/01/geo/wgs84_pos#
Description	Geographic longitude of an object
Domain	OGDObject
Data type	string

Many POIs have address or contact data attached. The vCard vocabulary (for which an RDF encoding [53] exists) seems very suitable for storing this kind of data. The selected properties from the vocabulary are tel, email and url for contact information.

Property	email
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The E-Mail adress of an object
Domain	OGDObject
Data type	string

Property	url
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The URL (internet adress) of an object
Domain	OGDObject
Data type	string

Property	tel
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The phone number of an object
Domain	OGDObject
Data type	string

For defining address data using the vCard vocabulary the properties country-name, region, postal-code and street-address were selected.

Property	country-name
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The name of the country an object lies within. In case of the Vienna OGD data the value of this property is always ”Austria”
Domain	OGDObject
Data type	string

Property	region
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The name of the region an object lies within. In case of the Vienna OGD data the value of this property is always ”Wien”
Domain	OGDObject
Data type	string

Property	postal-code
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The postal code of the region an object lies within.
Domain	OGDObject
Data type	string

Property	street-address
Type	Data Property
Namespace	http://www.w3.org/2006/vcard/ns#
Description	The street address of an object, containing the street name and house number but not the postal code or region.
Domain	OGDObject
Data type	string

In addition to localization information, the Vienna OGD POIs contain a lot of useful information like opening hours, localization hints or general information. Therefore the ontology defines some new properties that are used to store this information.

Property	localization_hint
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Localization hints, e.g. where to look for a certain object
Domain	OGDObject
Data type	string

Property	information
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Any other general information available for a POI object
Domain	OGDObject
Data type	string

Property	hrtype
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	A type identifier that is formatted for displaying in applications (in German language)
Domain	OGDObject
Data type	string

Property	opening_hours
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Any information about opening hours if applicable
Domain	OGDObject
Data type	string

Metadata properties

There are some metadata properties that have to be added to conform to requirements. All open data from the Vienna OGD platform is released using the ”Creative Commons Attribution 3.0” license [9]. According to the license terms, users are free to share and adapt the open data in a commercial or non-commercial way but have to attribute the work and refer to the city government of Vienna as data source. Therefore a source property has to be added. The Dublin Code vocabulary contains such a property.

Property	source
Type	Data Property
Namespace	http://purl.org/dc/elements/1.1/
Description	The source of the data
Domain	OGDObject, District
Data type	string

The Vienna OGD homepage clarifies the licensing terms and states that the term ”Datenquelle: Stadt Wien - data.wien.gv.at” has to be attached if the data is used. Therefore all individuals created will have this reference attached. It is also recommended to include a license for all data published as LOD, therefore the same license (”Creative Commons Attribution 3.0” [9]) is added to all individuals by using the license property from the dcterms vocabulary (an extension of the Dublin Core vocabulary). In addition the publisher property from the Dublin Core vocabulary is attached to the individuals.

Property	license
Type	Data Property
Namespace	http://purl.org/dc/terms/
Description	The license under which the data is shared
Domain	OGDObject, District
Data type	string

Property	publisher
Type	Data Property
Namespace	http://purl.org/dc/elements/1.1/
Description	The publisher of the converted datasets
Domain	OGDObject, District
Data type	string

Properties defining the demographic district data

Common vocabulary for Linked Open Data does not contain properties for defining demographic values of certain regions or cities. Therefore the properties for these information values are defined within the ontology.

Property	population
Type	Data Property
Namespace	—
Description	Total population of the district.
Domain	District
Data type	float

Property	district_population_ratio
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	The population ratio of a single district when compared to the city population
Domain	District
Data type	float

Property	population_density
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	The population density of a single district
Domain	District
Data type	float

Property	rate_female
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Rate of female population living in the district compared to the total population of the district.
Domain	District
Data type	float

Property	rate_male
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Rate of male population living in the district compared to the total population of the district.
Domain	District
Data type	float

Property	rate_foreign
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Rate of foreign population living in the district compared to the total population of the district.
Domain	District
Data type	float

Property	rate_migration
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Rate of people with migration background living in the district compared to the total population of the district.
Domain	District
Data type	float

Property	rate_youth
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Rate of young people (age <= 19 years) living in the district compared to the total population of the district.
Domain	District
Data type	float

Property	rate_retirees
Type	Data Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	Rate of elder people (age >= 65 years) living in the district compared to the total population of the district.
Domain	District
Data type	float

Object properties

Object properties are required to establish internal and external links. OWL contains the property owl:sameAs that defines that individuals are the same. This property is a good choice to express similarity with (seemingly distinct) resources from other data sources and therefore establish external links. Many linked data sources use this property to link to other data sources like DBpedia. In the use case of the Vienna OGD conversion, this property will be used to establish external links to DBpedia for elements where resources exist (e.g. districts, museums, parks).

Property	sameAs
Type	Object Property
Namespace	http://www.w3.org/2002/07/owl#
Description	States that individuals or resources are the same and is used to establish external links
Domain	OGDObject, District
Range	owl:Thing

The ontology also defines a property for internal links. Vienna OGD contains some internal links although they are not explicitly stated. The already mentioned example are playgrounds. In Vienna, many playgrounds are located within a park. The playground dataset contains the ”standort” attribute, but there is no stable link using e.g. the IDs. Such links will be detected during processing the data and be established using the lies_within property. This property will also be used to state that OGDObject individuals are located within a district individual.

Property	lies_within
Type	Object Property
Namespace	http://ogd.ifs.tuwien.ac.at/vienna
Description	States that an object lies within a district or another object
Domain	OGDObject
Range	OGDObject, District

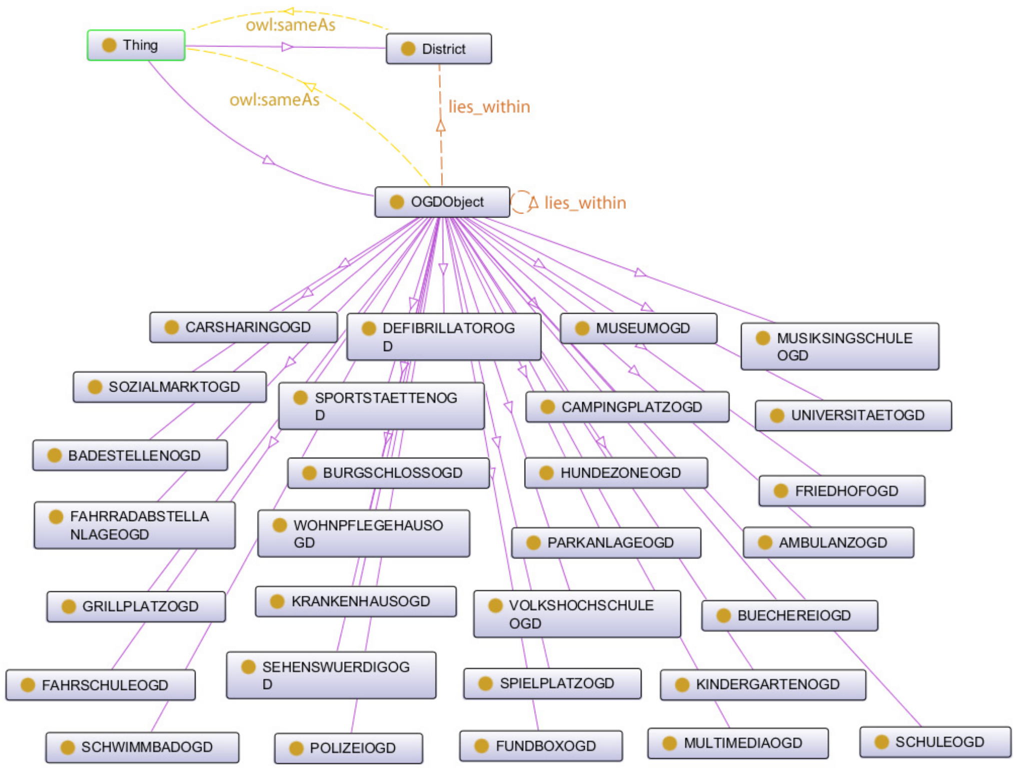


Figure 4.3: A visual representation of the Vienna OGD ontology

Figure 4.3 shows the ontology including classes and object properties.

Namespaces used

Due to the properties used before, the following namespaces are defined and used within the ontology:

Prefix	Namespace
(default namespace)	http://ogd.ifs.tuwien.ac.at/vienna
dc	http://purl.org/dc/elements/1.1/
rdfs	http://www.w3.org/2000/01/rdf-schema#
geo	http://www.w3.org/2003/01/geo/wgs84_pos#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
vcard	http://www.w3.org/2006/vcard/ns#
dcterms	http://purl.org/dc/terms/

4.6 Selecting technologies and frameworks

XML-based formats like RDF can be directly handled and processed by using XSLT (Extensible Stylesheet Language Transformation). XSLT was designed for this kind of transformation purpose. However, the option to work with an XSLT-based framework was rejected due to the nature of the project. There is a high amount of processing that needs the inclusion of specific algorithms and connected data sources, therefore a transformation using XSLT is not really an option.

It was decided to create a custom application using the Java programming language. The resulting application is object-oriented and platform independent.

Since CSV was chosen as the input data format, the opencsv library [24] was selected and included to handle processing of these files. The Apache log4j library [26] was selected and included for logging. Since the target format of the conversion process has to be RDF, it makes sense to rely on a framework for generating and handling RDF data.

4.7 Selecting an RDF framework

The most important part of the conversion process is the generation and manipulation of an RDF graph. Due to the expected size and complexity of the resulting model it is necessary to use or implement a framework for handling this model within the application. Two frameworks were considered and evaluated for this task:

1. Apache Jena [20]

2. OpenRDF.org Sesame [45]

Jena considers itself a ”Java framework for building Semantic Web applications” [20], Sesame considers itself an ”Java framework for storage and querying of RDF data” [45]. Both frameworks feature an extensive Java API and are available as open source applications. The Sesame framework is centered around its triplestore and repository functionality while the Jena framework allows to work with RDF models without creating such a repository.

Evaluation of Sesame

Sesame requires the creation of a local repository that contains and manages the RDF data. However, the creation of such a repository can easily be achieved:

```
LocalService service = Sesame.getService();
LocalRepository newRepository = service.createRepository(
"repositoryName", true);
```

In the case of this example the repository is stored in memory and only available while the application is running. However, since Sesame seems to be highly focused on storage functionality there are other alternatives. Sesame uses a layer called SAIL (Storage And Inference Layer) to allow for distinct but easily exchangeable ways of storing RDF data. Using this layer it is easy to access other options for storing data. One option is to use a native repository. A native repository stores its data directly to a file. The data format is binary and optimized for performance and size. Therefore it is not easily readable for humans. However, Sesame can also store repository data to a relational database. Maybe the most interesting option is to use a remote triplestore to store and work with data. In this case data can be distributed over several remote sources, which closely follows the Semantic Web vision.

Working with RDF data is also relatively simple when using the Sesame API. An RDF graph can be constructed by using the following code:

```
Graph newGraph = new org.openrdf.model.impl.GraphImpl();
newRepository.addGraph(newGraph);
```

Sesame uses its RDF Model API to work with RDF data. A statement can be generated by using the following commands:

```
URI doc2 = factory.createURI("http://example.org/document2");
URI creator = factory.createURI(
"http://purl.org/dc/elements/1.1/creator");
Literal creatorName = factory.createLiteral("Christian");
Statement newStatement = factory.createStatement(
doc2, creator, creatorName);
```

These lines create the RDF example from chapter 2 within Sesame.

Sesame supports SPARQL and remote querying of SPARQL endpoints. It is possible to directly retrieve data from sources like DBpedia by using the Sesame framework. Sesame features reasoning support and it is possible to plug in reasoners that support OWL reasoning.

Evaluation of Jena

Jena doesn't require the application to create or use a repository to work with RDF data. An object of type "Model" is used to store RDF triples. Such a model can be generated by using a static method of the ModelFactory class:

```
Model model = ModelFactory.createDefaultModel();
```

After creating a model, it can be populated by inserting statements or resources. Resources and properties can be directly created, again using the RDF example from chapter 2:

```
String doc2uri = factory.createURI("http://example.org/document2");
String creatorName = "Christian";
Resource doc2 = model.createResource(doc2uri);
doc2.addProperty(DC.creator, creatorName);
```

As an alternative to this approach, Jena also allows to create statements similar to the way Sesame does. It is possible to achieve the same result by using the following code:

```
String doc2uri = factory.createURI("http://example.org/document2");
String creatorName = "Christian";
Resource doc2 = model.createResource(doc2uri);
Statement s = model.createStatement(doc2 , DC.creator, creatorName );
model.add(s);
```

After the model was created it can be serialized to an RDF file. Jena also supports to import these kind of RDF files. Jena can use the RDF/XML serialization syntax but also supports other serialization formats like N3.

OWL is directly supported by Jena and the framework also supports SPARQL including remote querying of SPARQL endpoints. It is therefore possible to directly retrieve data from sources like DBpedia. Jena includes reasoning support and it is possible to use reasoners that support OWL reasoning.

Comparison and evaluation result

The most important feature required for the implementation is the construction and manipulation of an RDF graph. Both frameworks support this and allow to handle RDF data in a graph-based way. Another requirement is support of the SPARQL query language and an option to use remote SPARQL endpoints (like DBpedia). Again, both frameworks support this feature.

Performance of the frameworks was not evaluated since the size of data worked with is relatively small and the duration of the conversion process is not really an issue. It is difficult to find unbiased reports on performance regarding the two frameworks, however a report published by the Simile project evaluating the triplestore performance of the frameworks illustrates that it is difficult to pick a clear winner [21]. Each framework has advantages and disadvantages according to the storage type chosen.

Jena was chosen as a framework for processing RDF data during the conversion process. Although Sesame offers similar functionality and performance, Jena seems to be the more active framework when it comes to support, documentation and material as of today. A triplestore will not be used during the conversion process but only afterwards (to make the generated data remotely accessible and enable SPARQL querying).

4.8 System Architecture

The system is divided into of the following modules as seen in figure 4.4:

Source Data Extraction

The first module handles the retrieval of data from the Vienna OGD server. This is done within the Source Data Extractor that downloads the CSV files from the server and converts them into a data structure that can be used within the application to read out the data. The list of source files that are to be used is extracted from an XML configuration file. Some kind of cache is included in this module: The source data extractor is able to download and locally store the source files. Therefore it can be configured to directly use the downloaded files. This is intended primarily

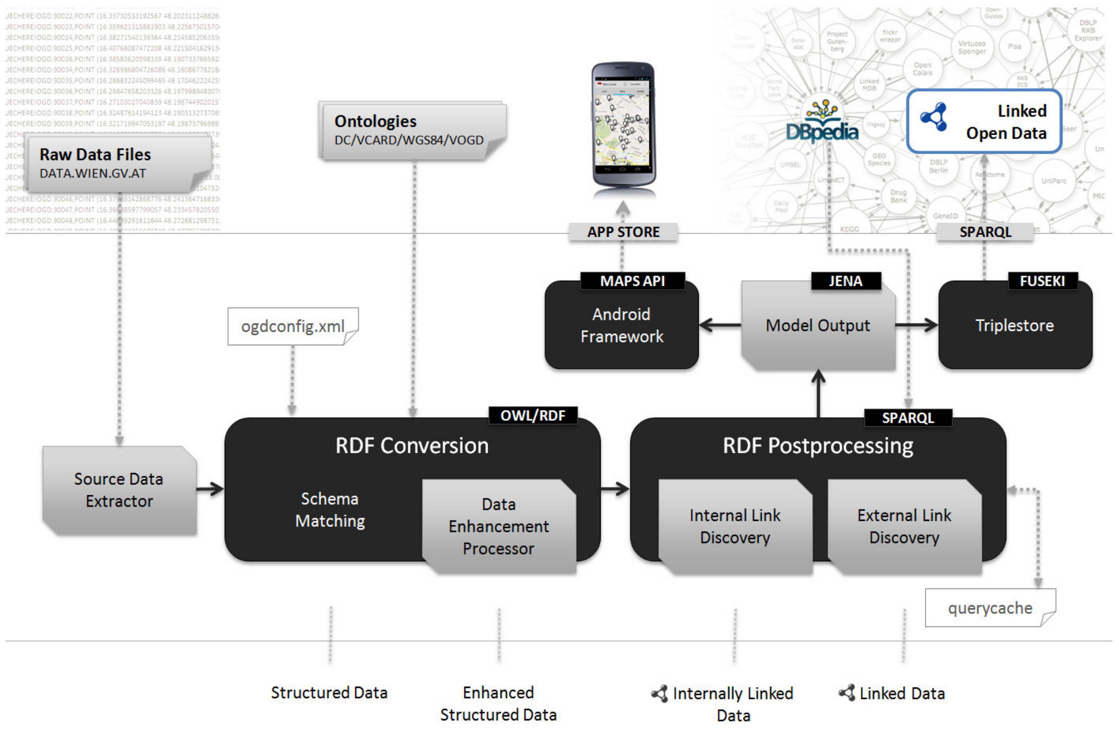


Figure 4.4: The system architecture of the application

for testing and developing purposes because processing is faster when offline files are used. The opencsv framework is then used to convert the CSV contents into native Java data structures that can be accessed when the data is converted into RDF data.

Data Enhancement Processor

The data enhancement processing module of the application improves the extracted data (now stored in Java objects) and prepares it for the RDF conversion. This is an important step because when the RDF data is generated only the properties defined by the ontology earlier will be used and the original source data doesn't fit into this structure. Therefore a data enhancement procedure is executed for every single dataset where patterns are found and analyzed to extract and prepare the data properties that are needed for a successful conversion.

Schema Matching

After the enhancement is finished the resulting data can be directly converted into RDF data. The Jena framework is utilized to do so. An empty model (GDF graph) is created and individuals as well as properties are inserted into this model. Jena natively supports the creation of properties from certain vocabularies like RDFS or Dublin Core by providing object-oriented representations. Such representations were manually added to the application for the Vienna OGD ontology created earlier or other vocabularies that are not natively included like WGS84. After this step a preliminary RDF graph exists in memory.

RDF Postprocessing

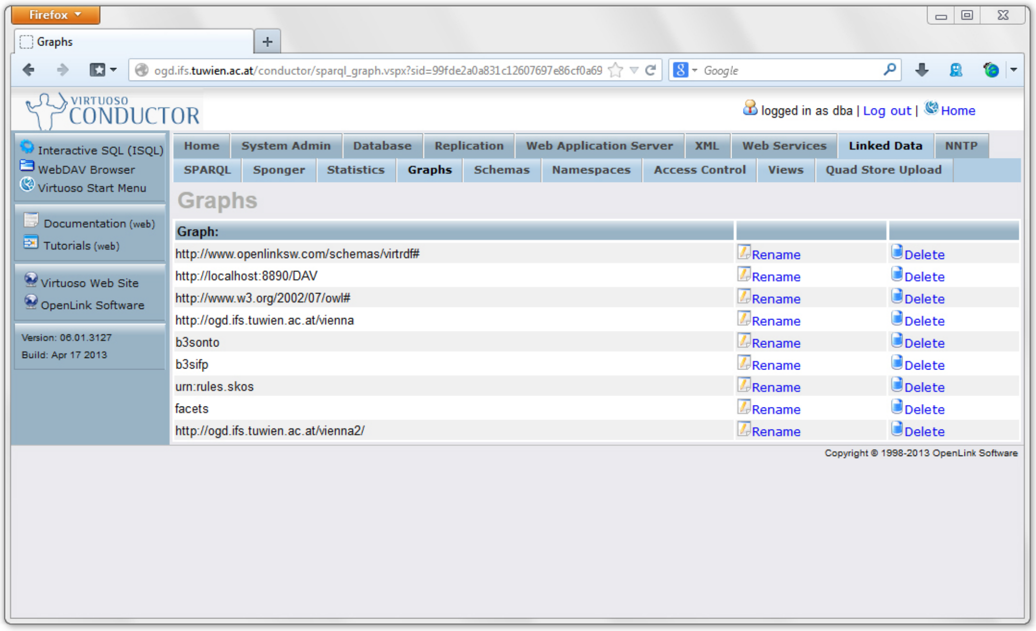
In this module, object properties are added that represent internal and external links. Since the RDF graph now already exists in memory it is possible to query the data using SPARQL. This option is utilized for creating the internal links. Those are created between:

- playgrounds and parks
- dog zones and parks
- any POI objects and districts

The creation of external links takes more effort. First, a series of SPARQL queries is executed after connecting to the DBpedia SPARQL endpoint. These queries are designed to retrieve labels and URIs of several resources within Vienna. The query results are written to a file for caching. Since this data doesn't change often and executing a series of SPARQL queries on a remote server can be a time-consuming process caching can speed up the execution. Data retrieved from DBpedia is then matched against the labels within the existing RDF graph by applying the Levenshtein string matching algorithm [23]. The tolerance is very low for this use case so only very close matches are accepted. If a match is found the applications adds a property of type owl:sameAs to link the individual to the matched DBpedia resource.

Model Output

Now that the model is completed and all properties exist in memory the RDF graph is written to a file. This functionality is natively supported by the Jena framework. The output format is XML-serialized RDF. In addition, a CSV file is generated that contains a flattened representation of individuals and can be used to quickly check the results or to transfer it to other applications.



Graphs

Graph:		
http://www.openlinksw.com/schemas/virtrdf#	Rename	Delete
http://localhost:8890/DAV	Rename	Delete
http://www.w3.org/2002/07/owl#	Rename	Delete
http://ogd.ifs.tuwien.ac.at/vienna	Rename	Delete
b3sonto	Rename	Delete
b3sifp	Rename	Delete
urn:rules.skos	Rename	Delete
facets	Rename	Delete
http://ogd.ifs.tuwien.ac.at/vienna2/	Rename	Delete

Figure 4.5: Virtuoso conductor interface showing a list of RDF graphs

4.9 Publishing the data

The resulting RDF data needs to be published on the web. OpenLink Virtuoso Open Source Edition [15] was chosen as a triplestore to achieve this. Virtuoso offers extensive web server capabilities that are needed for Linked Open Data use cases. A notable feature is the creation of views on relational databases to publish its content as Linked Open Data.

Virtuoso was chosen because of its powerful web server features and its sophisticated SPARQL functionality. Virtuoso describes the SPARQL features in more detail [15]:

> OpenLink Virtuoso supports SPARQL embedded into SQL for querying RDF data stored in Virtuoso's database. SPARQL benefits from low-level support in the engine itself, such as SPARQL-aware type-casting rules and a dedicated IRI data type. This is the newest and fastest developing area in Virtuoso.

Virtuoso features an extensive user interface (”conductor”) that was used to configure the server and upload the RDF graph. The user interface is depicted in Figure 4.5.

The SPARQL endpoint can be reached at:

`http://ogd.ifs.tuwien.ac.at/sparql`

4.10 Availability in a mobile app

Android [16] is an operating system designed for mobile phones and tablets that is based on Linux [22]. It was originally developed by a startup (Android Inc.) that was acquired by Google in 2005. The Android operating system is released under the Apache open source license and its full source code is available to developers. Android apps are applications that can be run on devices built upon the Android system.

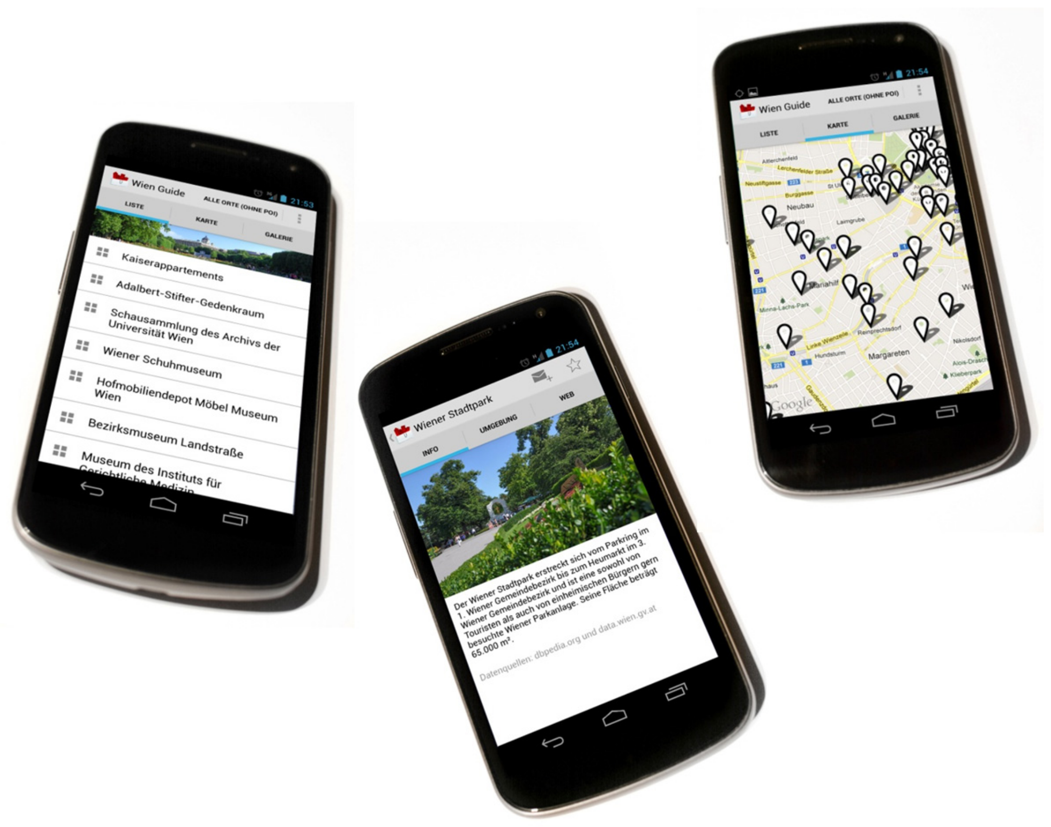
Reasons for releasing the app

In addition to making the data available via a SPARQL endpoint an Android app was created that visualizes the POI data. Vienna LOD mainly consists of POI data and a very common use case for such datasets is to display the POIs on a mobile phone or tablet. More than 10,000 POIs are processed during the conversion and by creating the app it is easier to present this data in a way that makes sense for end users. The resulting app is available through the Google Play Store [1]. Since the Vienna OGD source data is available in German language only the German version of the app (”Wien Guide”) features this data. Since the Android development SDK makes it easy to generate localized versions of apps, an English version is available but it features only data from DBpedia.

Structure of the app

The ViennaGuide app main activity features a list view of POIs. Every category that is converted is also included in the app. The user can select a category and has 3 different views available (selectable via tabs). The POIs can be shown in a list view, on a map or in a list view displaying only images (if available). When a POI is selected from any of the views the application switches to an activity showing more detail about it. Again, three views (selectable via tabs) are available. The user can display more information about the selected item (e.g. address or contact data), a map view or a web view. The web view displays the website associated with the item or - if none is available - results from a Google web search for the item. The user can bookmark POIs. Figure 4.6 shows several views of the user interface of the ViennaGuide app.

Android apps include a manifest file where system permissions the app requires have to be defined [37]. Users that install an app on Android have to confirm these permissions in order to be able to launch the application. Examples for such permissions are access to the users contacts or calendar, sending messages or accessing voicemail. The ViennaGuide app requires the following permissions:



Der Wiener Stadtpark erstreckt sich vom Parkring im 1. Wiener Gemeindebezirk bis zum Heumarkt im 3. Wiener Gemeindebezirk und ist eine sowohl von Touristen als auch von einheimischen Bürgern gern besuchte Wiener Parkanlage. Seine Fläche beträgt 65.000 m².

Datenquellen: dbpedia.org und data.wien.gv.at

Figure 4.6: The Android app featuring Vienna OGD

INTERNET Allows the application to access the Internet. This is required because the map has to be loaded via the web. Access to websites from within the app also relies on this permission.

ACCESS_FINE_LOCATION Allows the application to get the geographical location of the device. This is used for displaying the users current location on the map.

The manifest file is also used to define other components of the app, e.g. to register activities.

Data Handling within the app

When designing a mobile app, requirements differ from applications that are developed to run on a workstation or server. Since mobile devices have very limited processing power and a relatively low amount of memory available, handling an RDF graph with more than 190,000 triples may result in slow execution and response times. Some categories in the Vienna OGD dataset contain more than 500 items (e.g. there are more than 600 schools). Displaying that many objects on a mobile phone takes several seconds (e.g. building the map view) even if the items are already in memory. Loading them from a triple store or maybe even using a remote web service or SPARQL endpoint would reduce usability beyond an acceptable level on less powerful phones or if a slower Internet connection is used. Therefore all data is included with the app and can also be used offline to some extent (displaying items on a map is not possible without an Internet connection). To transfer the data to the application a CSV file is used. In this case, CSV was chosen because it is a very compact format that drastically reduces memory requirements for the app. The content of the CSV file is loaded into memory when the app is started.

CHAPTER 5

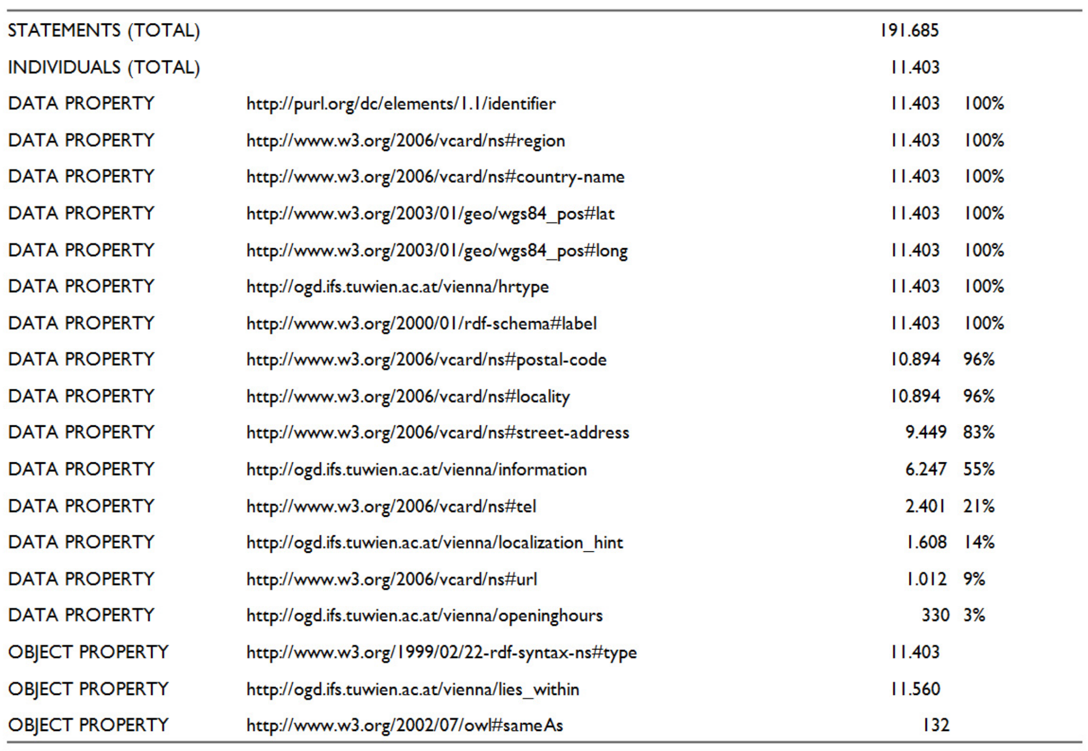
Results and Evaluation

5.1 Resulting RDF Graph

The result of the conversion process is an RDF graph containing 191,685 statements. Figure 5.1 lists a predicate count for the graph. There are a number of predicates that are attached to every single POI individual (e.g. ID, label, geographic coordinates, region) which is an improvement to the initial datasets where only ID and geographic position were attached to every single POI object. There are more than 100 external links (all of which point to DBpedia resources) and more than 600 internal links of type ”lies_within” between POIs. If districts are also taken into account there are more than 11,000 of these links because almost every POI individual is linked to a district individual.

An XML-serialized RDF representation of a POI individual (in this case of type ”SPIELPLATZOGD”) looks like this:

```
<rdf:Description
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD.58942">
<vcard:region>Wien</vcard:region>
<vogd:lies_within
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/PARKANLAGEOGD.58942"/>
<geo:long>16.386154487197135</geo:long>
<geo:lat>48.17240960988946</geo:lat>
<vcard:locality>Wien/Favoriten</vcard:locality>
<vogd:lies_within
```

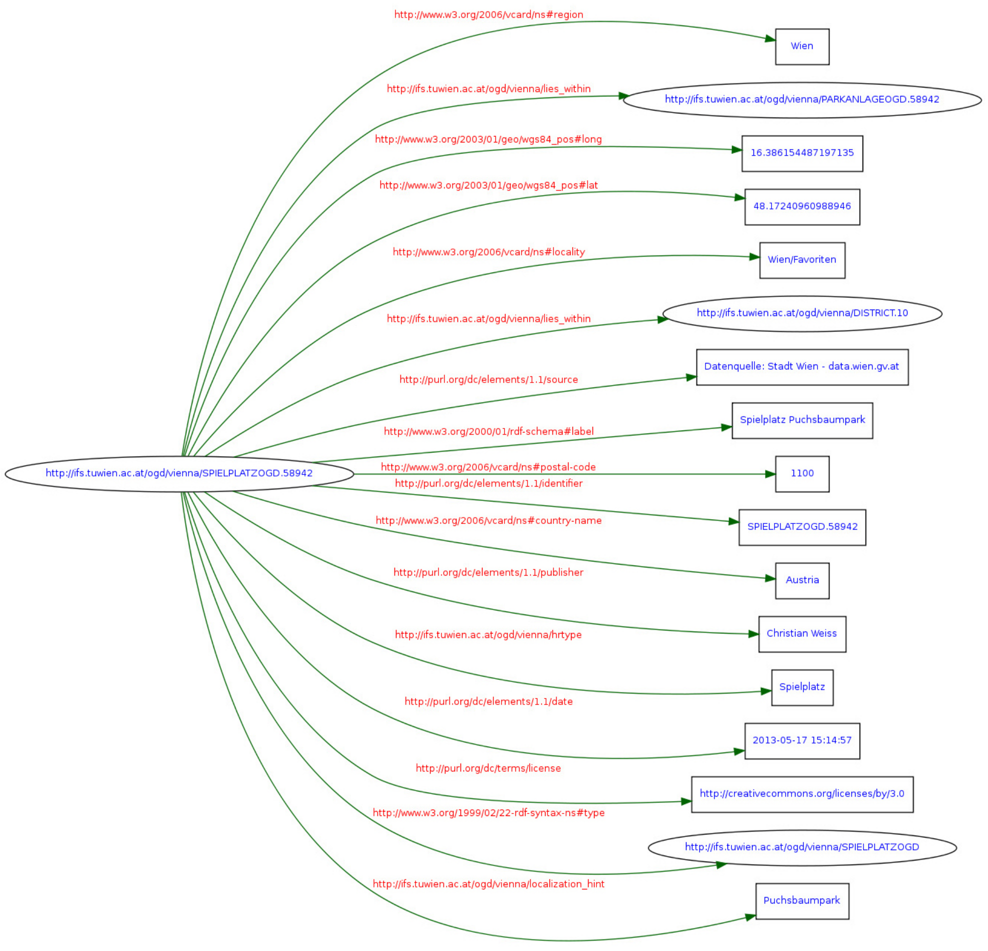


STATEMENTS (TOTAL)		191.685	
INDIVIDUALS (TOTAL)		11.403	
DATA PROPERTY	http://purl.org/dc/elements/1.1/identifier	11.403	100%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#region	11.403	100%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#country-name	11.403	100%
DATA PROPERTY	http://www.w3.org/2003/01/geo/wgs84_pos#lat	11.403	100%
DATA PROPERTY	http://www.w3.org/2003/01/geo/wgs84_pos#long	11.403	100%
DATA PROPERTY	http://ogd.ifs.tuwien.ac.at/vienna/hrtype	11.403	100%
DATA PROPERTY	http://www.w3.org/2000/01/rdf-schema#label	11.403	100%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#postal-code	10.894	96%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#locality	10.894	96%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#street-address	9.449	83%
DATA PROPERTY	http://ogd.ifs.tuwien.ac.at/vienna/information	6.247	55%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#tel	2.401	21%
DATA PROPERTY	http://ogd.ifs.tuwien.ac.at/vienna/localization_hint	1.608	14%
DATA PROPERTY	http://www.w3.org/2006/vcard/ns#url	1.012	9%
DATA PROPERTY	http://ogd.ifs.tuwien.ac.at/vienna/openinghours	330	3%
OBJECT PROPERTY	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	11.403	
OBJECT PROPERTY	http://ogd.ifs.tuwien.ac.at/vienna/lies_within	11.560	
OBJECT PROPERTY	http://www.w3.org/2002/07/owl#sameAs	132	

Figure 5.1: A count of statements in the resulting RDF graph

```
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/DISTRICT.10"/>
<dc:source>Datenquelle: Stadt Wien - data.wien.gv.at</dc:source>
<rdfs:label>Spielplatz Puchsbaumpark</rdfs:label>
<vcard:postal-code>1100</vcard:postal-code>
<dc:identifier>SPIELPLATZOGD.58942</dc:identifier>
<vcard:country-name>Austria</vcard:country-name>
<dc:publisher>Christian Weiss</dc:publisher>
<vogd:hrtype>Spielplatz</vogd:hrtype>
<dc:date>2013-05-17 15:14:57</dc:date>
<dcterms:license>http://creativecommons.org/licenses/by/3.0
</dcterms:license>
<rdf:type
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD"/>
<vogd:localization_hint>Puchsbaumpark</vogd:localization_hint>
</rdf:Description>
```

Figure 5.2 shows a graphical representation of this individual. Figure 5.3 extends this representation with connected individuals.



http://www.w3.org/2006/vcard/ns#region
Wien
http://ifs.tuwien.ac.at/ogd/vienna/lies_within
http://ifs.tuwien.ac.at/ogd/vienna/PARKANLAGEOGD.58942
http://www.w3.org/2003/01/geo/wgs84_pos#long
16.386154487197135
http://www.w3.org/2003/01/geo/wgs84_pos#lat
48.17240960988946
http://www.w3.org/2006/vcard/ns#locality
Wien/Favoriten
http://ifs.tuwien.ac.at/ogd/vienna/lies_within
http://ifs.tuwien.ac.at/ogd/vienna/DISTRICT.10
http://purl.org/dc/elements/1.1/source
Datenquelle: Stadt Wien - data.wien.gv.at
http://www.w3.org/2000/01/rdf-schema#label
Spielplatz Puchsbaumpark
http://ifs.tuwien.ac.at/ogd/vienna/SPIELPLATZOGD.58942
http://www.w3.org/2006/vcard/ns#postal-code
1100
http://purl.org/dc/elements/1.1/identifier
SPIELPLATZOGD.58942
http://www.w3.org/2006/vcard/ns#country-name
Austria
http://purl.org/dc/elements/1.1/publisher
Christian Weiss
http://ifs.tuwien.ac.at/ogd/vienna/hrtype
Spielplatz
http://purl.org/dc/elements/1.1/date
2013-05-17 15:14:57
http://purl.org/dc/terms/license
http://creativecommons.org/licenses/by/3.0
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://ifs.tuwien.ac.at/ogd/vienna/SPIELPLATZOGD
http://ifs.tuwien.ac.at/ogd/vienna/localization_hint
Puchsbaumpark

Figure 5.2: A visual representation of a POI individual

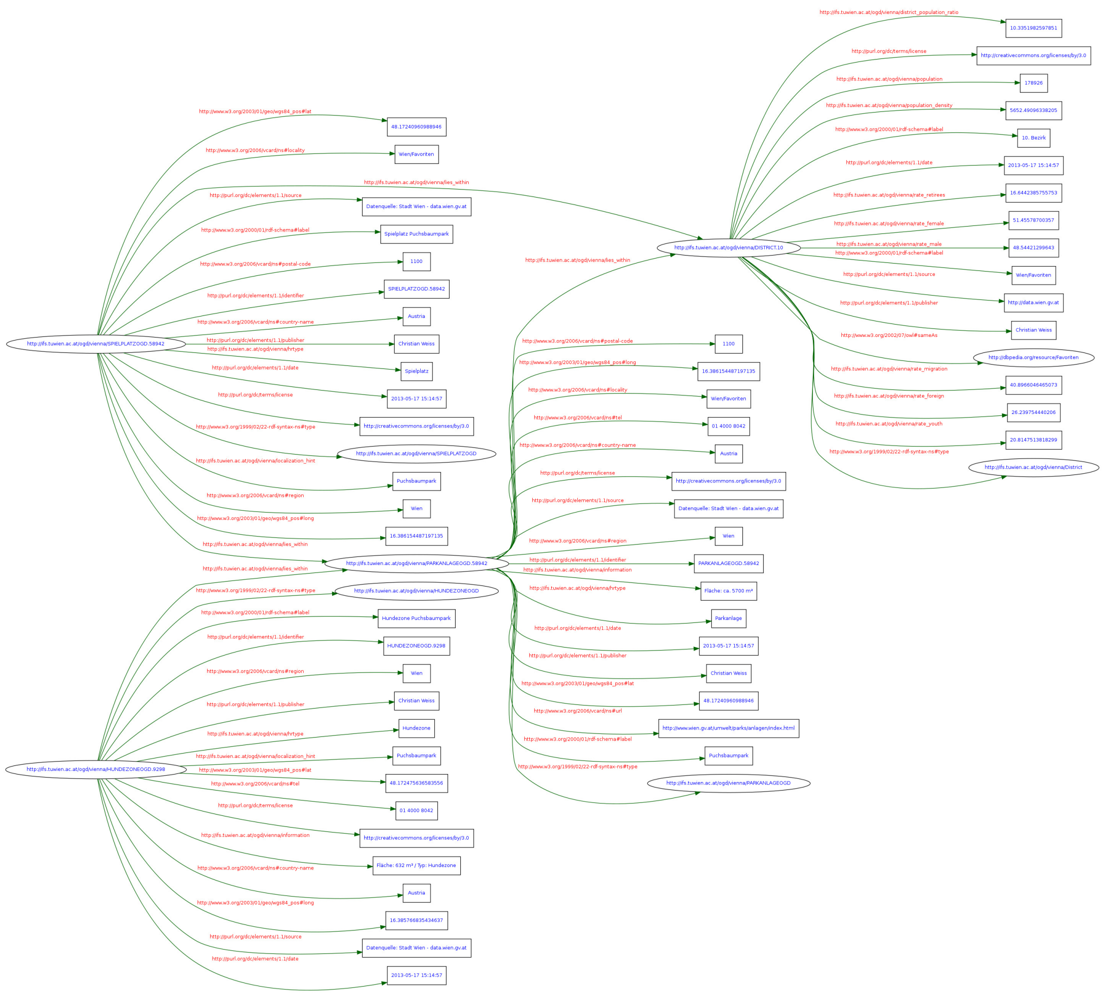


Figure 5.3: A visual representation of a POI in context with other individuals

Note that the "lies_within" property is defined as a transitive property in the ontology (see appendix). Since the playground individual is linked to a park individual and the park individual is linked to a district individual, the link connecting the playground individual to the district individual could be omitted if a reasoner is used.

Figure 5.4 shows a graphical representation of this RDF graph. Red squares denote individuals that are defined and described within the resulting RDF dataset while blue squares denote external resources. The graph was simplified for this representation: it was converted to an undirected graph and only a random subset of individuals was used to avoid cluttering. The remaining graph contains about 60,000 triples. External resources are located near the boundaries of the graph. The local individuals (red squares) near the boundaries that have a high degree are the district individuals that maintain a larger number of links to other individuals. Shown within the graph are only object properties of type ”lies_within” and ”owl:sameAs”. It is clearly visible that there are a lot of interconnections in this graph although links to remote resources are a minority when compared to internal links.

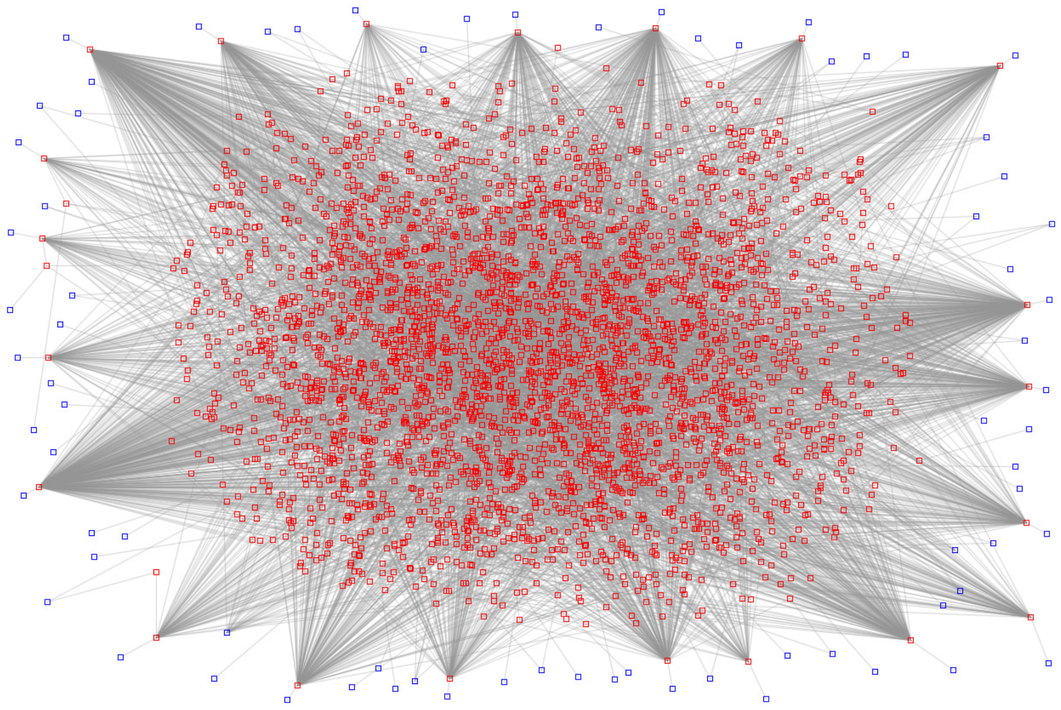


Figure 5.4: A graphical representation of the resulting RDF graph

Figure 5.5 shows another representation of the RDF graph. As before, red squares denote individuals that are defined and described within the resulting RDF dataset and blue squares denote external resources. This time the graph remains directional due to the smaller subset. Only individuals of the following types are included in this representation:

- http://ogd.ifs.tuwien.ac.at/vienna/PARKANLAGEOGD
- http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD
- http://ogd.ifs.tuwien.ac.at/vienna/HUNDEZONEOGD
- http://ogd.ifs.tuwien.ac.at/vienna/District

- External resources linked to individuals of the types above

However, all individuals of these types are included within the graph. This representation makes it easier to see links between some POIs (e.g. near the lower boundary of the graph).

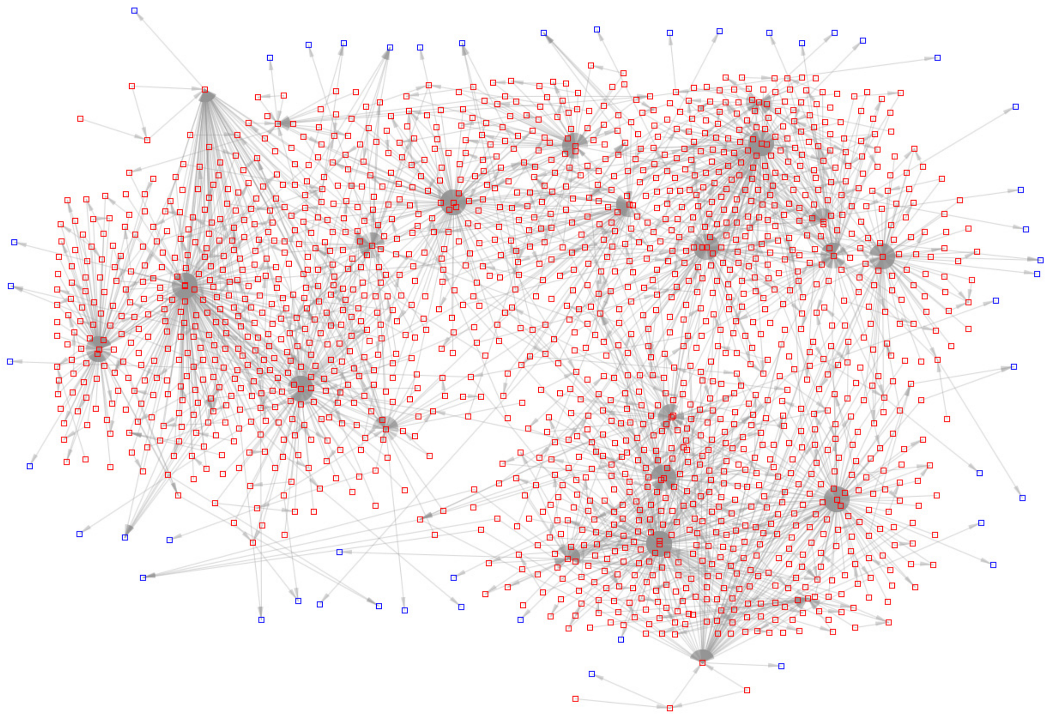


Figure 5.5: A subset of the graph focusing on lies_within predicates

5.2 Query Examples

With some basic knowledge of SPARQL (described in chapter 2) it is easy to formulate some queries on the RDF graph. Two examples will be listed here.

Query Example 1

The first SPARQL query will list the geographic position of parks (latitude/longitude) that are located in districts where less than 15% of the population are older than 65 years and feature a dog zone as well as a playground.

```
SELECT DISTINCT ?parklabel ?rate_retirees ?lat ?long
WHERE {
 ?park <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 <http://ogd.ifs.tuwien.ac.at/vienna/PARKANLAGEOGD>.
 ?park <http://www.w3.org/2000/01/rdf-schema#label> ?parklabel.
 ?park <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?lat.
 ?park <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?long.
 ?playground <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 <http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD>.
 ?playground <http://ogd.ifs.tuwien.ac.at/vienna/lies_within>
 ?park .
 ?dogzone <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 <http://ogd.ifs.tuwien.ac.at/vienna/HUNDEZONEOGD>.
 ?dogzone <http://ogd.ifs.tuwien.ac.at/vienna/lies_within> ?park.
 ?park <http://ogd.ifs.tuwien.ac.at/vienna/lies_within>
 ?district.
 ?district <http://ogd.ifs.tuwien.ac.at/vienna/rate_retirees>
 ?rate_retirees .
 FILTER (?rate_retirees  < 15).
} ORDER BY ASC(?parklabel)
```

These are the results of the first query example:

parklabel	rate_retirees	lat	long
Alfred-Grünwald-Park	14.93	48.198	16.362
Auer-Welsbach-Park	13.23	48.187	16.318
Bacherpark	13.79	48.188	16.356
Braunhuberpark	14.27	48.171	16.415
Dadlerpark	13.23	48.190	16.329
Einsiedlerpark	13.79	48.186	16.350
Ernst-Lichtblau-Park	13.79	48.183	16.351
Forschneritschpark	13.23	48.197	16.319
Herderpark	14.27	48.172	16.410
Hyblerpark	14.27	48.182	16.413
Ingeborg-Bachmann-Park	14.99	48.258	16.444
Klieberpark	13.79	48.185	16.363
Luise-Montag-Park	14.27	48.168	16.416
Otto-Affenzeller-Park	14.99	48.232	16.450
Parkanlage Am Hundsturm	13.79	48.187	16.347
Parkanlage Leopold-Rister-Gasse	13.79	48.181	16.355

Parkanlage Margaretengürtel	13.79	48.185	16.345
Reithofferpark	13.23	48.198	16.331
Rohrauerpark	13.23	48.206	16.315
Rudolf-Sallinger-Park	13.79	48.188	16.363
Schloss Neugebäude - Unterer Garten	14.27	48.161	16.443
Teich Hirschstetten	14.99	48.241	16.479
Vogelweidpark	13.23	48.204	16.332
Weghuberpark	13.96	48.206	16.356

Query Example 2

The second SPARQL query will list all playgrounds that are located in districts where more than 20% of the population are aged below 19 years. The playgrounds have to be located within a public park for which a DBpedia resource (and therefore a Wikipedia.org article) exists.

```
SELECT DISTINCT ?playground ?districtlabel ?rateyouth
WHERE {
 ?object <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 <http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD>.
 ?object <http://ogd.ifs.tuwien.ac.at/vienna/lies_within>
 ?park.
 ?object <http://www.w3.org/2000/01/rdf-schema#label>
 ?playground.
 ?park <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 <http://ogd.ifs.tuwien.ac.at/vienna/PARKANLAGEOGD>.
 ?park <http://www.w3.org/2002/07/owl#sameAs> ?wiki.
 ?park <http://ogd.ifs.tuwien.ac.at/vienna/lies_within>
 ?district.
 ?district <http://www.w3.org/2000/01/rdf-schema#label>
 ?districtlabel.
 ?district <http://ogd.ifs.tuwien.ac.at/vienna/rate_youth>
 ?rateyouth.
 FILTER (?rateyouth > 20).
}
```

These are the results of the second query example:

playground	**districtlabel**	**rateyouth**
Spielplatz Herderpark	Wien/Simmering	22.11
Spielplatz Donaupark	Wien/Donaustadt	22.78
Spielplatz Draschepark	Wien/Liesing	20.01
Spielplatz Haydnpark	Wien/Meidling	20.10
Spielplatz Wilhelmsdorfer Park	Wien/Meidling	20.10
Spielplatz Waldmüllerpark	Wien/Favoriten	20.81
Spielplatz Blumengärten Hirschstetten	Wien/Donaustadt	22.78

5.3 Result Evaluation

Before implementing the conversion process a list of 10 requirements were defined (see Chapter 4). This list can now be reviewed and evaluated.

1. The data has to be available on the web with an open license

   The data was published on the web and is publicly reachable via the SPARQL endpoint. The data was released using the ”Creative Commons Attribution 3.0” license [9], the same license that was used for the source data.

2. It has to be available as machine-readable structured data in a non-proprietary format

   The data is available in XML-serialized RDF format. RDF is a format defined by the W3C for machine-readable information [41].

3. Open standards from the W3C have to used

   RDF [41] and OWL [36] are open standards defined by the W3C. They were used to define the structure and publish the data on the web.

4. The data has to be linked to other data sources

   The original converted dataset features 132 links to external sources. All of them are links to DBpedia resources (75 unique).

5. The resulting dataset should consist of at least 100,000 triples including metadata

   There are 196,685 triples in the dataset. A breakdown of predicates is listed earlier in this chapter.

6. The aforementioned uniformity issues should be resolved

   Several measures were taken to improve data quality. All POIs that contain any kind of address data have that data converted to the standardized vCard format. As mentioned, there is a number of predicates that are now attached to every POI individual (label, geographic coordinates, region, etc.). The Vienna OGD datasets used as source have only ID and geographic position attached to every single POI object.

7. There should be internal links (object properties between resources) in addition to the external links

   The implementation of the conversion process generates internal links between certain kinds of POIs. The property used for these links is ”lies_within”. The links are visible in the visualizations earlier in this chapter. The SPARQL query

   ```
   SELECT ?a ?b WHERE {
   ?a <http://ogd.ifs.tuwien.ac.at/vienna/lies_within> ?b.
   }
   ```

   reveals all individuals that were connected.

8. The dataset should follow a common model in form of an OWL ontology

   An OWL ontology was introduced to define the vocabulary. The ontology is listed in the appendix to this document.

9. Existing vocabulary should be used for properties if possible

   Vocabulary from the following standards/ontologies was used to define properties:

Vocabulary	Location
Dublin Core (DC)	http://purl.org/dc/elements/1.1/
RDF Schema	http://www.w3.org/2000/01/rdf-schema#
WGS 84	http://www.w3.org/2003/01/geo/wgs84_pos#
OWL	http://www.w3.org/2002/07/owl#
RDF	http://www.w3.org/1999/02/22-rdf-syntax-ns#
vCARD	http://www.w3.org/2006/vcard/ns#
DC Terms	http://purl.org/dc/terms/

10. The data is published on a server so it can be queried using SPARQL

    The SPARQL endpoint can be publicly accessed at

    `http://ogd.ifs.tuwien.ac.at/sparql`

The defined requirements could be fulfilled by the proposed solution. It was possible to convert the existing data sources into LOD using existing schemata. The data was published on the web and is now available as machine-readable structured data applying W3C standards as well as an open license and therefore qualifies as Linked Open Data according to guidelines and requirements [19].

CHAPTER 6

Future work

While a significant part of the available Vienna OGD data was used for the LOD transformation there is still data that was not included in the conversion process yet. This is because the focus of this thesis was to show how a conversion could be done and not to convert all available data. Data that was left out and could be converted in the future includes statistical data and POIs that do not have a point location but a path or area location. Examples for such objects are hiking paths or bicycle routes, both available at the Vienna OGD platform. Transferring more demographic or statistical data would also increase options for formulating more complex queries.

It is also possible to add data from other cities into the Linked Open Data cloud by using or enhancing the ontology established in this thesis. For example, The Austrian city of Linz also publishes Open Government Data. Some of the datasets are quite similar to what Vienna OGD publishes, however they again use mostly distinct schemata. There is some coordination anyway, especially regarding establishment of a common metadata standard [34].

It is possible to increase the degree of interlinking by either adding manual links (this only makes sense when already considered with the source data design) or by using more hints to automatically establish such links. For example, it would be possible to detect related POIs not only by comparing their labels but also by detecting closeness due to their geographic position (longitude/latitude and/or address data). Knowing the district boundaries that are also available from the Vienna OGD platform, it would also be possible to assign a property linking to the corresponding district individual to POIs that don't have any address data attached. This would require geospatial processing.

The processing engine developed as part of this thesis is fault tolerant to some degree with regards to the source data but not all cases of possible errors or glitches within the source data were taken into account. Therefore the conversion process needs some monitoring. A possible extension would be to develop automated execution monitoring and integrity checks for the resulting data. This monitoring framework could then send automated reports in case there are any problems with the source data. Another option for improvement would be to make transformation rules (for example handling of non-uniformity between address properties of POIs) configurable via a configuration file. This would enable the setup of an update process without the need to edit source code. If the structure of a source file changes, the current system cannot adapt automatically and either the source code has to be adapted or the source file needs to be excluded from processing (this can currently be done using a configuration file). Another possible extension would be versioning support to reflect changes over time and corrected or retracted data.

Finally, more in-depth analysis or visualization of the resulting data was not part of the thesis. Specifically, aside from the app that was used to visualize some of the resulting data, there is no way to visually browse the resulting dataset. Triplestores like Sesame offer some sort of data browsing, however a custom application that provides a user interface enabling to specifically browse the resulting Vienna Linked Open Data would make the results easier to survey for people who not want to use SPARQL queries to retrieve results. Similar to what was done by data.gov.uk it would also be a possible extension to create an API for developers.

CHAPTER 7

Conclusion

The Semantic Web Vision introduces significant improvements to the World Wide Web. However, the gap between traditional web technologies and Semantic Web technologies is very high, therefore slowing down its introduction. The concepts outlining the Semantic Web are already around for 15 years - a very long time frame when it comes to web development - yet, the implementation of these concepts on a broader basis is still lacking. There are only few Semantic Web applications around because data that can be used is scarce. There is not much data encoded in Semantic Web based formats like RDF around because too few applications exist that catalyze the generation of linked data and offer incentive to do so. The emergence of Linked Open Data platforms, most notable the DBpedia database has improved that situation. There are billions of triples available that can be processed and used by Semantic Web applications. The DBpedia dataset has become very successful and is used by many applications around the world in one way or the other.

The Semantic Web vision is a top-down concept. However, innovation in web technologies and structures was often a bottom-up process, driven by communities or users. This was demonstrated by the so-called Web 2.0 that changed the web toward social communities and user-generated content. Linked Open Data works like a bottom-up approach in several ways. It can be seen as a cloud of data that dynamically grows as new linked data is added. Not all data added to the Linked Open Data cloud can be considered to be highly useful as of today. There are datasets that might never be used in real-world applications because they exist purely for academic or demonstration purposes. However, government agencies around the world maintain very large sets of data that would be very useful for real-world applications. Vienna OGD is a very good example. Despite being released in isolated sets using somehow diverse schemata, already more than 80 applications were developed by an active developer community that utilize this data. Adding this data to the Linked Data Cloud can be a valuable contribution toward Semantic Web development.

This thesis has shown that a conversion to true 5-star rated Linked Open Data can be achieved with the heterogeneous data that is provided by the city government of Vienna without redesigning the source data. This approach is more similar to what was done in the United States by the Data-gov Wiki. In the United Kingdom on the other hand, the open data initiative was designed to support Linked Open Data from the ground up. The latter approach might prove to be more sustainable on the long term.

During the conversion there were challenges that had to be resolved to guarantee compatibility with LOD requirements and guidelines. Vienna OGD is neither linked internally nor externally. This was to be expected of course, however links to other data sources are a very important requirement when it comes to LOD. In this thesis, such links were generated by comparing labels of individuals by applying a string metric algorithm. The resulting quantity and quality of links are sufficient to fulfill established LOD standards. However, many more links could be included if they are already considered when generating the data. Since Vienna OGD is generated from a variety of sources and therefore does not use a homogeneous schema for all datasets a direct 1:1 conversion into LOD while re-using existing schemata was not an option. This thesis has shown that the existing data can be modified, improved and converted using existing schemata like Dublin Core or vCard without redesigning the source data. The resulting homogeneous dataset follows LOD guidelines and requirements. It is available via a public SPARQL endpoint that enables processing of complex queries that couldn't be executed on the source data alone.

CHAPTER 8

Appendix

8.1 Acronyms

CSV	Comma Separated Values
DL	Description Logics
FOAF	Friend of a Friend
GIS	Geographic Information System
GML	Geography Markup Language
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LOD	Linked Open Data
OGD	Open Government Data

OWL	Web Ontology Language
POI	Point of Interest
RDF	Resource Description Framework
RDFS	RDF Schema
SAIL	Storage And Inference Layer
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema
XSL	Extensible Stylesheet Language
XSLT	XSL Transformation

8.2 OGD Ontology

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
    <!ENTITY dcterms "http://purl.org/dc/terms/" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
    <!ENTITY vcard "http://www.w3.org/2006/vcard/ns#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY geo "http://www.w3.org/2003/01/geo/wgs84_pos#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
xml:base="http://www.w3.org/2002/07/owl"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:vcard="http://www.w3.org/2006/vcard/ns#"
xmlns:dcterms="http://purl.org/dc/terms/">
<Ontology rdf:about="http://ogd.ifs.tuwien.ac.at/vienna">
<rdfs:comment xml:lang="en">Vienna OGD Ontology used to
transform data from Open Government Data Wien (OGD;
http://data.wien.gv.at) into Linked Open Data as part
of the Master Thesis of Christian Weiss at Vienna
University of Technology</rdfs:comment>
</Ontology>

<!--
//////////////////////////////////////////////////////////////
//
// Object Properties
//
//////////////////////////////////////////////////////////////
-->

<!-- http://ogd.ifs.tuwien.ac.at/vienna/lies_within -->

<ObjectProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/lies_within">
<rdf:type rdf:resource="&owl;TransitiveProperty"/>
<rdfs:range
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
```

```
<rdfs:range
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</ObjectProperty>

<!-- http://www.w3.org/2002/07/owl#sameAs -->

<ObjectProperty rdf:about="&owl;sameAs">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</ObjectProperty>

<!--
///////////////////////////////////////////////////////////////
//
// Data properties
//
///////////////////////////////////////////////////////////////
-->

<!-- http://ogd.ifs.tuwien.ac.at/vienna/district_population_ratio -->

<DatatypeProperty
rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/district_population_ratio">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/hrtype -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/hrtype">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range
 rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/information -->
```

```
<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/information">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range
 rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/localization_hint -->

<DatatypeProperty
rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/localization_hint">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range
 rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/openinghours -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/openinghours">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:rang
e rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/population -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/population">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/population_density -->
```

```
<DatatypeProperty
rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/population_density">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/rate_female -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/rate_female">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/rate_foreign -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/rate_foreign">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/rate_male -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/rate_male">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/rate_migration -->
```

```
<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/rate_migration">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/rate_retirees -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/rate_retirees">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/rate_youth -->

<DatatypeProperty
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/rate_youth">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:range
 rdf:resource="&xsd;float"/>
</DatatypeProperty>

<!-- http://purl.org/dc/elements/1.1/date -->

<DatatypeProperty rdf:about="&dc;date">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</DatatypeProperty>

<!-- http://purl.org/dc/elements/1.1/identifier -->

<DatatypeProperty rdf:about="&dc;identifier">
```

```
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range
 rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://purl.org/dc/elements/1.1/publisher -->

<DatatypeProperty rdf:about="&dc;publisher">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</DatatypeProperty>

<!-- http://purl.org/dc/elements/1.1/source -->

<DatatypeProperty rdf:about="&dc;source">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</DatatypeProperty>

<!-- http://purl.org/dc/elements/1.1/type -->

<DatatypeProperty rdf:about="&dc;type">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://purl.org/dc/terms/license -->

<DatatypeProperty rdf:about="&dcterms;license">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</DatatypeProperty>
```

```
<!-- http://www.w3.org/2000/01/rdf-schema#label -->

<DatatypeProperty rdf:about="&rdfs;label">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/District"/>
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</DatatypeProperty>

<!-- http://www.w3.org/2003/01/geo/wgs84_pos#lat -->

<DatatypeProperty rdf:about="&geo;lat">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2003/01/geo/wgs84_pos#long -->

<DatatypeProperty rdf:about="&geo;long">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#country-name -->

<DatatypeProperty rdf:about="&vcard;country-name">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#email -->

<DatatypeProperty rdf:about="&vcard;email">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
```

```
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#locality -->

<DatatypeProperty rdf:about="&vcard;locality">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#postal-code -->

<DatatypeProperty rdf:about="&vcard;postal-code">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#region -->

<DatatypeProperty rdf:about="&vcard;region">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#street-address -->

<DatatypeProperty rdf:about="&vcard;street-address">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#tel -->

<DatatypeProperty rdf:about="&vcard;tel">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
```

```
</DatatypeProperty>

<!-- http://www.w3.org/2006/vcard/ns#url -->

<DatatypeProperty rdf:about="&vcard;url">
<rdfs:domain
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
<rdfs:range rdf:resource="&xsd;string"/>
</DatatypeProperty>

<!--
//////////////////////////////////////////////////////////////
//
// Classes
//
//////////////////////////////////////////////////////////////
-->

<!-- http://ogd.ifs.tuwien.ac.at/vienna/AMBULANZOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/AMBULANZOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/BADESTELLENOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/BADESTELLENOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/BUECHEREIOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/BUECHEREIOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/BURGSCHLOSSOGD -->
```

```
<Class
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/BURGSCHLOSSOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/CAMPINGPLATZOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/CAMPINGPLATZOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/CARSHARINGOGD -->

<Class
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/CARSHARINGOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/DEFIBRILLATOROGD -->

<Class
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/DEFIBRILLATOROGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/District -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/District"/>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/FAHRRADABSTELLANLAGEOGD -->

<Class
 rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/FAHRRADABSTELLANLAGEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
```

```
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/FAHRSCHULEOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/FAHRSCHULEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/FRIEDHOFOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/FRIEDHOFOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/FUNDBOXOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/FUNDBOXOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/GRILLPLATZOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/GRILLPLATZOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/HUNDEZONEOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/HUNDEZONEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/KINDERGARTENOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/KINDERGARTENOGD">
```

```
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/KRANKENHAUSOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/KRANKENHAUSOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/MULTIMEDIAOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/MULTIMEDIAOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/MUSEUMOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/MUSEUMOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/MUSIKSINGSCHULEOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/MUSIKSINGSCHULEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/OGDObject -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject">
<rdfs:subClassOf>
<Restriction>
<onProperty rdf:resource="&geo;lat"/>
<qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1
</qualifiedCardinality>
```

```
<onDataRange rdf:resource="&xsd;string"/>
</Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<Restriction>
<onProperty rdf:resource="&geo;long"/>
<qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1
</qualifiedCardinality>
<onDataRange rdf:resource="&xsd;string"/>
</Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<Restriction>
<onProperty rdf:resource="&dc;identifier"/>
<qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1
</qualifiedCardinality>
<onDataRange rdf:resource="&xsd;string"/>
</Restriction>
</rdfs:subClassOf>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/PARKANLAGEOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/PARKANLAGEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/POLIZEIOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/POLIZEIOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/SCHULEOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SCHULEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
```

```
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/SCHWIMMBADOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SCHWIMMBADOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/SEHENSWUERDIGOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SEHENSWUERDIGOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/SOZIALMARKTOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SOZIALMARKTOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SPIELPLATZOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/SPORTSTAETTENOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/SPORTSTAETTENOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/UNIVERSITAETOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/UNIVERSITAETOGD">
```

```
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/VOLKSHOCHSCHULEOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/VOLKSHOCHSCHULEOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>

<!-- http://ogd.ifs.tuwien.ac.at/vienna/WOHNPFLEGEHAUSOGD -->

<Class rdf:about="http://ogd.ifs.tuwien.ac.at/vienna/WOHNPFLEGEHAUSOGD">
<rdfs:subClassOf
 rdf:resource="http://ogd.ifs.tuwien.ac.at/vienna/OGDObject"/>
</Class>
</rdf:RDF>
```

Bibliography

[1] Vienna Guide app. https://play.google.com/store/apps/details?id=candroid.viennaguide, 2013. Last accessed: 2013-05-15.

[2] Soeren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, , and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *The semantic web*, pages 722–735. Springer, 2007.

[3] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. 2001.

[4] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[5] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked Data on the Web. 2008.

[6] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[7] World Wide Web Consortium: RDF Concepts. http://www.w3.org/tr/rdf-concepts/, 2004. Last accessed: 2013-05-13.

[8] World Wide Web Consortium Linked Data Cookbook. http://www.w3.org/2011/gld/wiki/linked_data_cookbook, 2011. Last accessed: 2013-05-13.

[9] creativecommons.org. http://creativecommons.org/licenses/by/3.0/legalcode, 2013. Last accessed: 2013-05-13.

[10] Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/, 2006. Last accessed: 2013-05-13.

[11] UK Government Linked Data. http://data.gov.uk/linked-data, 2013. Last accessed: 2013-05-19.

[12] About DBPedia. http://dbpedia.org/about, 2013. Last accessed: 2013-05-15.

[13] Li Ding, Dominic DiFranzo, Alvaro Graves, James R Michaelis, Xian Li, Deborah L McGuinness, and Jim Hendler. Data-gov wiki: Towards linking government data. *BCH+].(Cit. on p.)*, 2010.

[14] Li Ding, Lina Zhou, Tim Finin, and Anupam Joshi. How the semantic web is being used: An analysis of foaf documents. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 113c–113c. IEEE, 2005.

[15] OpenLink Software: Virtuoso Open Source Edition. http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/main/, 2013. Last accessed: 2013-05-15.

[16] Android for Developers. http://developer.android.com, 2013. Last accessed: 2013-05-15.

[17] Johann Höchtl, Martin Kaltenböck, Peter Parycek, Judith Schossböck, and Thomas Thurner. Open government data: Potentiale, risiken und hürden. *Abrufbar unter: www. user. tu-berlin. de/komm/CD/paper/061121. pdf*, 2011.

[18] Dublin Core Metadata Initiative. http://dublincore.org/, 2013. Last accessed: 2013-05-13.

[19] Tim Berners-Lee: Linked Data Design Issues. http://www.w3.org/designissues/linkeddata.html, 2006. Last accessed: 2013-05-13.

[20] Apache Jena. http://jena.apache.org, 2011. Last accessed: 2013-05-12.

[21] Ryan Lee. Scalability report on triple store applications. 2004.

[22] Wei-Meng Lee. *Beginning Android 4 Application Development*. Wrox, 2012.

[23] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[24] Open CSV Library. http://opencsv.sourceforge.net/, 2011. Last accessed: 2013-05-19.

[25] Open Government License. http://www.nationalarchives.gov.uk/doc/open-government-licence/, 2011. Last accessed: 2013-05-19.

[26] Apache log4j library. http://logging.apache.org/log4j/2.x/, 2013. Last accessed: 2013-05-19.

[27] Tim Berners-Lee: Semantic Web Road map. http://www.w3.org/designissues/semantic.html, 1998. Last accessed: 2013-05-13.

[28] Google Merchants. www.google.com/merchants/, 2013. Last accessed: 2013-05-13.

[29] Peter Mika and Tim Potter. Metadata statistics for a large web corpus. *LDOW*, 937, 2012.

[30] 8 Principles of Open Government Data. http://www.opengovdata.org/home/8principles, 2007. Last accessed: 2013-05-15.

[31] City Government of Vienna. http://data.wien.gv.at/, 2013. Last accessed: 2013-05-13.

[32] City of Vienna: OGD. http://data.wien.gv.at/neuigkeiten/wege/, 2013. Last accessed: 2013-05-13.

[33] United Kingdom Government Cabinet Office. Open source, open standards and re-use: government action plan. *https://www.gov.uk/government/publications/open-source-open-standards-and-re-use-government-action-plan*, 2010.

[34] Cooperation OGD. http://www.data.gv.at/hintergrund-infos/cooperation-ogd-oesterreich/, 2013. Last accessed: 2013-05-23.

[35] Wikipedia: Ontology. http://en.wikipedia.org/wiki/ontology, 2013. Last accessed: 2013-05-13.

[36] World Wide Web Consortium: OWL. http://www.w3.org/2004/owl/, 2004. Last accessed: 2013-05-13.

[37] Android Permissions. http://developer.android.com/reference/android/manifest.permission.html, 2013. Last accessed: 2013-05-15.

[38] Sunlight Foundation: 10 Principles. http://sunlightfoundation.com/policy/documents/ten-open-data-principles/, 2010. Last accessed: 2013-05-15.

[39] W3C LOD Project. http://esw.w3.org/topic/sweoig/taskforces/communityprojects/linkingopendata, 2007. Last accessed: 2013-05-15.

[40] Facebook Open Graph Protocol. http://ogp.me/, 2013. Last accessed: 2013-07-19.

[41] World Wide Web Consortium: RDF. http://www.w3.org/rdf/, 2004. Last accessed: 2013-05-13.

[42] World Wide Web Consortium: RDFa. http://www.w3.org/tr/xhtml-rdfa-primer/, 2012. Last accessed: 2013-05-13.

[43] World Wide Web Consortium: RDFS. http://www.w3.org/tr/rdf-schema/, 2004. Last accessed: 2013-05-13.

[44] schema.org. http://www.schema.org/, 2013. Last accessed: 2013-07-19.

[45] OpenRDF Sesame. http://www.openrdf.org, 2013. Last accessed: 2013-05-12.

[46] John Sheridan and Jeni Tennison. Linking uk government data. *BHBL+].: http://events. linkeddata. org/ldow2010/.(Cit. on p.)*, 2010.

[47] World Wide Web Consortium: SPARQL. http://www.w3.org/tr/rdf-sparql-query/, 2008. Last accessed: 2013-05-13.

[48] Linked Open Data Stack. http://www.opendataimpacts.net/2011/05/whats-in-the-linked-open-data-stack/, 2010. Last accessed: 2013-05-15.

[49] The Data Hub Statistics. http://datahub.io/stats, 2013. Last accessed: 2013-05-19.

[50] Thomas Steiner, Raphaël Troncy, and Michael Hausenblas. How google is using linked data today and vision for tomorrow. *Linked Data in the Future Internet*, 2010.

[51] W3C Introduction to SKOS. http://www.w3.org/2004/02/skos/intro, 2004. Last accessed: 2013-05-19.

[52] Data.gov.uk: URIs. http://data.gov.uk/resources/uris, 2013. Last accessed: 2013-05-19.

[53] World Wide Web Consortium: vCard. http://www.w3.org/submission/2010/subm-vcard-rdf-20100120/, 2010. Last accessed: 2013-05-13.

[54] J von Lucke and CP Geiger. Open government data-frei verfügbare daten des öffentlichen sektors (2010), 2010.

[55] w3schools.com. http://www.w3schools.com/html/html5_semantic_elements.asp, 2013. Last accessed: 2013-07-19.

[56] Tim Berners-Lee: Semantic Web. http://www.w3.org/2000/talks/1206-xml2k-tbl/slide10-0.html, 2000. Last accessed: 2013-05-13.

[57] Data.gov Wiki. http://data-gov.tw.rpi.edu/wiki, 2013. Last accessed: 2013-05-19.