Master's Thesis in Geographical Information Science No. 90

Transformation and linking of authoritative multi-scale geodata for the Semantic Web: A case study of Swedish national building data sets

Eiður Kristinn Eiðsson

2018
Department of
Physical Geography and Ecosystem Science /
Centre for Geographical Information Systems
Lund University
Sölvegatan 12
223 62 Lund
Sweden

Eiður Kristinn Eiðsson: *Transformation and linking of authoritative multi-scale geodata for the Semantic Web: A case study of Swedish national building data sets*.

Master's thesis, 30 credits, in Geographical Information Science. Department of Physical Geography and Ecosystem Science, Lund University. 2018.

Transformation and linking of authoritative multi-scale geodata for the Semantic Web: A case study of Swedish national building data sets

Eiður Kristinn Eiðsson

Master’s thesis, 30 credits, in Geographical Information Science

Supervisors

Lars Harrie

Weiming Huang

Examiners

Mahdi Farnaghi

David Tenenbaum

Department of Physical Geography and Ecosystem Science
Lund University
2018

Abstract

Semantic Web technology has attracted much interest for its ability to integrate and deliver information from different sources, and increasingly, geodata are being published as triple data according to the graph structure laid out in the Resource Description Framework (RDF). Although the Semantic Web already contains a large volume of geotagged information, there is still a lack of suitable geometric representations for proper visualization thereof. Data held by national mapping agencies and other authoritative bodies would therefore be an invaluable addition, since they are typically of high quality and often produced at multiple scales. This thesis presents a case study where the feasibility of bringing authoritative multi-scale geodata to the Semantic Web is investigated through a practical implementation involving nationally maintained Swedish building data sets. Three polygon-based products, each intended for display at a certain scale level, were transformed into RDF using a combination of GeoSPARQL and draft INSPIRE vocabularies. To test integration with other sources, the building data were linked with the community-maintained gazetteer GeoNames. The transformed and linked data were then uploaded to a triplestore, and from there, they were queried and visualized in a web map. The case study shows that the selected data sets can be brought to the Semantic Web for enhanced visualization of related information, but careful planning and harmonization of the original data should preferably precede the transformation process. Furthermore, they show that current geospatial vocabularies are largely suitable for the task, but will need some refinement and added capabilities before extensive data publication can commence.

Keywords: geographical information systems (GIS), Resource Description Framework (RDF), linked geodata, INSPIRE application schemas, data transformation, cartographic scale.

Table of contents

List of figures

List of abbreviations

CRS	Coordinate reference system
EC	European Commission
GI	Geographical information
GML	Geography Markup Language
GSD	*Geografiska Sverigedata*
INSPIRE	Infrastructure for Spatial Information in the European Community
JSON	JavaScript Object Notation
LoD	Level of detail
MRDB	Multiple representation database
NMA	National mapping agency
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
UML	Unified Modelling Language
URI	Uniform Resource Identifier
VGI	Volunteered geographical information
W3C	World Wide Web Consortium
WFS	Web Feature Service
WKT	Well-known Text
XML	Extensible Markup Language

1. Introduction

1.1. Background

The last two decades have seen much interest in publishing information as Linked Data on the Semantic Web, and practitioners of geographical information technology have been some of its early adopters (Janowicz et al., 2012). This is unsurprising, given that many kinds of data have a geographical component, and geographical information (GI) often acts as a common link between different data (Hart and Dolbear, 2013, ch.2). Geographical data sets published on the Semantic Web therefore play a pivotal interlinking role therein, as evidenced by their central, interwoven placements in the Linking Open Data cloud (Abele et al., 2017). For producers and users of geodata, Semantic Web technology offers much potential, especially regarding integration and accessibility (Kuhn, Kauppinen, and Janowicz, 2014).

The underlying technology centres around the Resource Description Framework (RDF), a specification by the World Wide Web Consortium (W3C, 2014a), where data are stored and linked as triples in a graph structure, and entities are identified by Uniform Resource Identifiers (URIs). It is accompanied by a number of serialization formats for reading and writing triple-based data. The data are encoded according to so called vocabularies, in which concepts and relationships are defined. The GI community has, in recent years, commenced the large task of developing shared (i.e. widely agreed upon) vocabularies for the geospatial domain, as well as software to implement them (Battle and Kolas, 2012). An important milestone in this effort is the GeoSPARQL standard (OGC, 2012a) which defines a basic geographical RDF data model and query language.

The realization of a Geospatial Semantic Web requires a substantial amount of geodata to be open, i.e. publicly available. Increasingly, national mapping agencies (NMAs) are releasing their data sets free of charge, both through direct downloads and on-demand services (Lantmäteriet, 2017a). National mapping agencies have begun showing interest in publishing their data in RDF and demonstrated the process in pilot projects, notably by the UK Ordnance Survey (Goodwin, Dolbear, and Hart, 2008), US Geological Survey (Usery and Varanka, 2012), and others. The availability of open data has also grown in response to the rise of volunteered geographical information (VGI), a phenomenon where geocoded information is produced by the users themselves (Goodchild, 2007). Likewise, such data have been subject to transformation into RDF, as in the LinkedGeoData project (Stadler et al., 2012).

The inherent ability of RDF to link data from different sources opens up opportunities to integrate a wealth of information containing some geographical component, or “rich information”, with the “rich geometries” maintained by producers of GI. While the Semantic Web has, and continues to be, populated with a great amount of geotagged information, the introduction of rich geometries has lagged behind. Although there have already been demonstrations by NMAs on transforming some of their data to RDF and publishing them online, this has not lead to a widespread adoption of the practice. To this day, only a few pioneering projects remain the only examples of such undertakings, representing just a fraction of the data themes and spatial coverage needed to realize a more enabled Geospatial Semantic Web.

The use of the technology within the realm of VGI has attracted interest due to the large volume of data involved (covering all countries) and relative ease of integration with other projects. While the proliferation of VGI greatly increases the wealth of available data, their quality, reliability, and source credibility inevitably come into question (Flanagin and Metzger, 2008). As volunteers capture data in different ways or derive them from different sources, their combined input is inconsistent and often undocumented. This causes the map scale, or rather, level of detail (LoD) of the resulting data to be heterogenous, which leads to problems in subsequent map production (Touya and Brando, 2013; Touya and Reimer, 2015).

This situation is somewhat different with authoritative data, i.e. data produced by high- and low-level government agencies, organizations, and professional enterprises. Those typically produce and publish data at given scales or LoDs depending on their intended use. Different representations of single geographical features are either maintained separately, or by conversion from one LoD to another using generalization techniques. The current direction is to maintain Multiple Representation Databases (MRDBs), in which features are represented by geometries at several resolution levels, stored together, and explicitly linked (Sarjakoski, 2007). A comparison has been made between the MRDB approach and presenting geodata in RDF (Hahmann and Burghardt, 2010). Whichever method is used to produce the original data, their delivery through the Semantic Web could benefit from both multiple representation of features and the consistency that authoritative data provide.

Early examples of transformations of geodata into RDF by NMAs and collaborative VGI projects mostly relied on ad hoc vocabularies, which accomplished their tasks but were not ideal for further integration. The GI community therefore embarked on the process of defining vocabularies that may be shared across related disciplines (Salas et al., 2011; Battle and Kolas, 2012). Designing vocabularies that fulfil the requirements of all users is a large and complex task (Hart and Dolbear, 2013, ch.10), and while there have been developments in recent years, as with the advent of GeoSPARQL, the activity remains a work in progress. If a large amount of data is to be published as Linked Data, it would be of much benefit to users at all stages (from data compilation to map production) to have them structured and linked using standard vocabularies in common use.

1.2. Aim

As described above, an increasing amount of geodata is being published on the Semantic Web, yet appropriate geometric representations for quality mapping are often lacking or not easily integrated with other data. Data sets produced by authoritative bodies like NMAs usually contain geometries that are of high accuracy, little heterogeneity, and often available at several levels of detail. For this reason, they have the potential to enrich the Semantic Web with quality data suitable for a variety of user applications. Meanwhile, shared geospatial RDF vocabularies are being developed to make their transformation and linking possible. With these observations in mind, the aim of this thesis is to investigate whether authoritative geodata at multiple levels of detail can be effectively brought to the Semantic Web given the technical situation of today. For a successful achievement of such a task, one can identify certain interdependent requisites (Figure 1) that lead to the following research questions:

- Can multi-scale national data sets be transformed and linked for the Semantic Web to enhance visualization of related data therein?
- Do geospatial vocabularies currently support the transformation of national data sets at multiple scales?

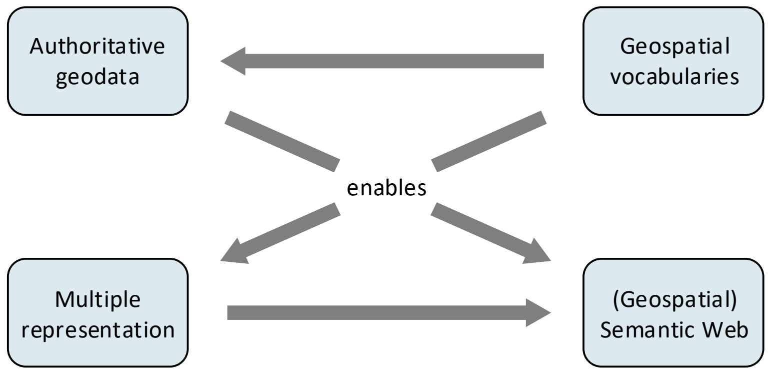


Figure 1. Requisites for bringing authoritative geodata at multiple levels of detail to the Semantic Web.

1.3. Study design

To assess the current situation, a practical implementation involving Swedish building data sets is presented as a case study. The buildings theme is one that commonly appears in web maps, an application area that may benefit from multiple representation geodata delivered using Semantic Web technology. The opportunities of linked geodata have attracted interest in Sweden, which relates to wider aspirations regarding reusability of open government data. An investigative project on the topic has already been carried out by a group of government agencies and academics (Östman and Blomqvist, 2014), but practical effort towards publication of data sets has so far been limited.

In the case study, three polygon data products, maintained at the national level and published in a traditional file format, are transformed into RDF and stored. At present, the most appropriate way to encode these data is to use two closely intertwined vocabularies, one being that of GeoSPARQL, an international standard for describing and querying geographical objects in RDF, and the other being draft vocabularies based on the European INSPIRE application schemas for describing objects within a variety of geographical themes. To demonstrate how authoritative data can supplement other sources of information, the transformed multi-scale building data are linked to GeoNames, a community-maintained gazetteer that supports data retrieval in RDF form.

Figure 2 shows the main stages of the case study. After necessary preparations, the three authoritative building data sets undergo a transformation process where their records are extracted, linked to each other, and written to an output document according to the rules laid out in the vocabularies. The geometries are annotated with LoD information, which is given in the form of cartographic scale levels (representative fractions) recommended by the data producer. This requires the design of a custom scale vocabulary, since no alternative currently exists. Another process then links the GeoNames gazetteer data with the authoritative building data, the output of which is stored in a triplestore (graph database) along with the transformed authoritative data and the relevant vocabularies. Finally, three use cases are presented in which the stored data are queried and visualized, demonstrating some (but not all) of their capabilities with respect to multiple representation of features.

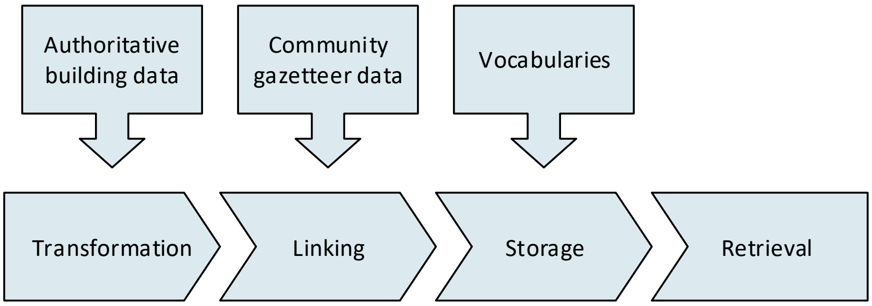


Figure 2. Main stages of the case study.

1.4. Disposition

The technical framework underlying the previously stated aim is introduced in Chapter 2, with concepts relating to the Semantic Web and RDF coming first, and those behind multiple representation second. It ends with a summary of literature dealing with transformation of authoritative geodata into RDF. The materials used in the caste study are presented in Chapter 3, beginning with a demarcation of a reasonably sized study area. Next, the structure of the three national building data sets is described in detail, followed by that of the community gazetteer data used for testing data integration. The shared RDF vocabularies used to encode and link the building data are then described in some detail. The methods for transforming and linking these data using said vocabularies are detailed in Chapter 4, and the output is briefly described in the beginning of Chapter 5. The remainder of Chapter 5 presents the results of the case study in the form of three selected use cases. The lessons that can be drawn from the implementation as a whole are discussed in Chapter 6, which is followed by short conclusions in Chapter 7.

2. Technical framework

2.1. Semantic Web

2.1.1. Resource Description Framework

The Semantic Web is seen as an extension of the current Web, where information has well-defined meaning, evolving from a Web of Documents to a Web of Data. An important aspect of this concept is what is called the Linked Data or Linked Open Data (LOD) paradigm, in which data should be linked and open to allow both humans and machines to more easily discover and make sense of information (Berners-Lee, 1998; 2006). The Semantic Web is enabled by a collection of technologies that together form the Resource Description Framework (RDF) from the W3C (2014a). It sets forth a node-edge graph structure in which resources are linked via a set of subject-predicate-object triples (two nodes with an edge between them).

Each part, or term, of a triple is a resource denoting a particular thing, or referent (a physical object, document, concept, relationship, value etc.), preferably identified by a Uniform Resource Identifier (URI). For example, the country of Sweden could, as a subject, be identified by the URI <http://example.org/SWE>. A predicate denotes a uniquely identified relationship definition such as <http://example.org/inContinent> or <http://example.org/hasName>. An object term may either be a URI, such as <http://example.org/EUR>, or a literal, which is a stored value, such as “Sweden”, of a given data type (a string in this case). Subjects and objects may also be blank nodes, which are locally scoped resource surrogates with local labels but no URIs. A combination of a subject, a predicate, and an object forms a triple (Figure 3), and a set of triples forms an interlinked RDF graph. With a graph in place, data can be retrieved based on a user-specified triple pattern using the SPARQL query language (W3C, 2013).

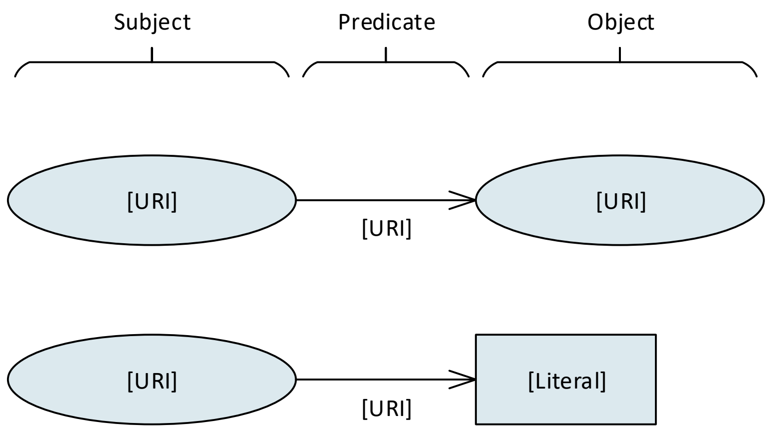


Figure 3. Depiction of two kinds of triples.

Resources and triples have no meaning on their own, but depend on formal definitions of concepts and relationships. Classes, properties, and rules of a given conceptual schema are codified in a collection of definitions, called a vocabulary. A vocabulary is therefore a representation of a specific domain’s ontology, which is a model of what things are, do, have etc., and the two terms are often used interchangeably. A concept modelled in a vocabulary can therefore be of a certain type, consist of parts, have attributes etc. For example, Sweden is a country, which is a type of administrative unit, which can consist of sub-units and have a name.

The advantage of describing data this way is that computer systems can apply artificial reasoning to make logical inferences from the data encoded with the vocabulary (Schreiber et al., 2014).

W3C (2014b) specifies RDFS, a fundamental data modelling vocabulary that can be extended to create further vocabularies. It provides, among other things, the classes *rdfs:Resource*, *rdfs:Class*, *rdfs:Literal*, *rdfs:DataType*, and *rdf:Property*. One class can inherit from another through the property *rdfs:subClassOf*. An instance is declared by linking its URI with a class through the property rdf:type. A property can inherit from another through the property *rdfs:subPropertyOf*. The classes that the subject and/or object of a predicate should belong to, its domain and range, can be stated with *rdfs:domain* and *rsfs:range* respectively. More advanced modelling vocabularies and rule languages exist, most notably OWL (W3C, 2004a). OWL extends RDFS with base classes *owl:Class* (a subclass of *rdfs:Class*), and *owl:ObjectProperty* and *owl:DatatypeProperty* (both subclasses of *rdf:Property*), whose instances differ in whether the object is an ordinary resource or a literal. It further provides properties and property types to describe logical relationships between classes (e.g. *owl:equivalentClass*), properties (e.g. *owl:equivalentProperty* and *owl:SymmetricProperty*), and instances (e.g. *owl:sameAs*). For literals, RDF vocabularies typically reuse and extend the base data types of XML Schema (W3C, 2004b), known by the prefix *xsd*.

A number of serialization formats for writing triple data are available, ranging from simple URI/literal lists to more human-readable ones. One common format is RDF/XML (W3C, 2014c) which, as the name suggests, is used to read and write RDF data in XML. Among other things, it allows the use of prefixes for repeated namespaces (e.g. *ex:has* as short for <http://example.org/has>) and nesting of predicates and objects belonging to the same subject. Another common format is Turtle (W3C, 2014d), which has a flexible, human-readable syntax with similarities to SPARQL. It allows resources to be written as either URIs or prefixed entities (much like RDF/XML), and nesting of both predicates and objects (using semicolons and commas). An instance of a class can be declared with the predicate “a”, which is a shorthand for *rdf:type*. The language of a string can be provided with the suffix “@”, and the data type of a literal is identified by the suffix “^^”. The examples below show RDF data written in these two formats. Both contain the same three triples: “Sweden is a country”, “Sweden is in Europe”, and “Sweden is named ‘Sweden’”.

RDF/XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ex="http://example.org/">
    <rdf:Description rdf:about="http://example.org/SWE">
        <rdf:type rdf:resource="http://example.org/country" />
        <ex:inContinent rdf:resource="http://example.org/EUR" />
        <ex:hasName>Sweden</ex:hasName>
    </rdf:Description>
</rdf:RDF>
```

Turtle:

```
@prefix ex: <http://example.org/> .

ex:SWE a ex:country ;
    ex:inContinent ex:EUR ;
    ex:hasName "Sweden"@en .
```

2.1.2. Geospatial Semantic Web

The opportunities of Semantic Web technology attracted interest from the GI community early on (Egenhofer, 2002), prompting research into the requirements of a Geospatial Semantic Web (Kolas, Hebeler, and Dean, 2005; Lieberman, 2006). Since then, many theoretical and technical aspects of this new field have been studied, including ontology formulation and alignment, querying capability, and software support to name a few, and new challenges have been identified (Janowicz et al., 2012; Battle and Kolas, 2012; Kuhn, Kauppinen, and Janowicz, 2014). The practical feasibility of transforming, linking, and publishing geodata in RDF has been the subject of several studies and projects in recent years, both concerning VGI data, like in the LinkedGeoData project (Stadler et al. 2012), and data held by public authorities (see Section 2.3).

The question of how to encode geographical data in RDF inevitably arose with the commencement of data transformations within the geospatial domain. The first studies on the matter were therefore partly devoted to vocabulary design, which lead to isolated project-specific solutions. Early collaborations to create shared geospatial vocabularies include the Basic Geo Vocabulary (Brickley, 2003), which is limited to points in WGS 84 (with altitude), the W3C Geospatial Vocabulary (Lieberman, Singh, and Goad, 2007), which includes GeoOWL and extends the first with more geometry classes, and the NeoGeo Vocabulary (Salas et al., 2011), which is based on GML/Simple Features. The need for a standard vocabulary for GI led to work in the OGC that culminated in GeoSPARQL (OGC, 2012a) and an accompanying Simple Features vocabulary, which are described further in Section 3.4.1.

While GeoSPARQL and its predecessors lay the foundation for encoding geographical information in RDF, they are (intentionally) restricted in their scope, dealing only with core aspects like coordinates, basic spatial classes, spatial relations etc. Geodata obviously involve further information in the form of attributes and metadata, and additional vocabularies are therefore needed to describe specific spatial concepts. One example is the Core Location Vocabulary (EC, 2012) which provides classes and properties for two themes, geographical names and addresses, and reuses the geometry serializations of the basic vocabularies. Metadata for spatial data sets and services can be encoded according to the GeoDCAT-AP profile (EC, 2016) which reuses properties from several common vocabularies. Currently, the detailed data models developed under the INSPIRE Directive are being mapped into RDF vocabularies (JRC, 2017b), which are described further in Section 3.4.2.

2.2. Multiple representation

2.2.1. Core concepts

Multiple representation refers to the ability to depict a real-world object on a map with a suitable geometry, selected from more than one possible abstraction. It is directly related to the process of generalization, an intrinsic aspect of cartography. The goal of generalization is to emphasize more important information and reduce clutter to make a map legible. Traditionally, cartographers selected relevant information to display according to scale and function, but this had to be done independently for each scale level. Computerized map-making has allowed maintenance of large-scale data that can then be reused in the production of small-scale maps, but this also presents new challenges. The process of generalization is often divided into model generalization, which deals with data classification and storage, and cartographic generalization, which is for optimizing visualization. The latter presents a set of main operations, known by terms such as simplification, aggregation, exaggeration, typification, displacement etc. (Mackaness and Chaudhry, 2008; Hardy and Field, 2012).

Kilpeläinen (2000) and Sarjakoski (2007) give a good overview of the field and describe the concept of a multiple representation database (MRDB). The production process might take place in two stages: the model generalization stage and the cartographic generalization stage. Data are maintained at a high level of detail, which serves as the base level in a multi-level database (the MRDB). The levels are linked together in one way or another, and model generalization operations are carried out on an object in one level to automatically maintain the same object in the next. A building object represented as a detailed polygon could, for example, be simplified to a less detailed polygon in one operation, and to a point in the next. The second stage involves cartographic generalization of the stored MRDB data to produce several map product databases. Hahmann and Burghardt (2010) compare MRDBs with Linked Data and note similarities in their approach to storing and linking several representations of the same phenomenon, but also apparent differences in their purpose, focus, and supervision.

The idea of an object's representation is inherently linked to the concepts of scale, resolution, and level of detail (LoD). Montello (2001) lists three principal meanings given to the term "scale": cartographic scale, analysis scale, and phenomenon scale. The relationship between the size of a phenomenon and its depicted size on a map is distinguished with the term "cartographic scale". It is commonly presented with the representative fraction, which is a numerical ratio between a distance on a map and a distance on earth (e.g. 1:50 000), but can also be presented as a verbal ratio or using a scale bar. Analysis scale refers to the unit size in which phenomena are measured and aggregated (such as the size of raster cells). Phenomenon scale is the scope of phenomena, e.g. "local scale" and "national scale". The term "resolution" (a kind of analysis scale) is usually defined along the lines of "the number of resolvable things per unit distance" (Hardy and Field, 2012), although it commonly overlaps with other meanings (Degbelo and Kuhn, 2012). Since digital data sets have no map distance, Goodchild and Proctor (1997) argue that the representative fraction is a legacy of earlier technology that is no longer suitable as the primary indicator of the more general concept of "level of detail".

2.2.2. Vocabulary support

There is currently no agreed upon way of handling such concepts as scale, resolution, or level of detail in RDF, and proposals are scarce. In previous studies where several levels have been included (Vilches et al., 2010a; 2010b; 2014; Kadaster, 2016; Debruyne et al., 2017) they appear to be only implicitly referenced within the name of each graph. The GeoDCAT-AP metadata profile (EC, 2016), which is used for describing spatial data sets and services as a whole, includes a provisional property for spatial resolution in the form of *rdfs:comment*. This is sufficient for describing uniform data sets, but might be too inflexible in some situations.

Annex B.2 of (OGC, 2012a) contains a demonstration of how the built-in vocabulary of GeoSPARQL might be reused in an application-specific scenario, and includes a way to represent features with multiple geometries. There, a new custom object property, being a sub-property of *geo:hasGeometry* (see Section 3.4.1), is defined for each representation level. This kind of solution means that one feature instance may be associated with several geometry instances via equally many distinctly-named predicates, such as *hasPointGeometry* or *hasScale10000*. Such an approach has been considered by Huang, Mansourian, and Harrie (2016) and utilized in a study by Xu (2017).

Carral et al. (2013) point out that the usability of digital map representations relies on creating "smarter data instead of smarter applications", but information on which scale a representation is compatible for tends to be hidden within the latter. They propose a scale ontology which aims to allow users to query the underlying scale information of data and services, and thus request data from different sources that are appropriate for the application. The model is not described in full here, but most importantly, a class *ScaledRepresentation* is a reified tuple which describes a class *GeometricRepresentation* as a function of classes *GeographicThing* and *ScaleLevel*. Further, *hasUpperBound* and *hasLowerBound* are introduced as properties of *ScaleLevel*.

While any one approach may well suit the needs of a given application, a shared vocabulary is preferable from an interoperability point of view. Multiple representation is one of the topics that have been brought up regarding publication and usage of spatial data on the Web (Tandy, Brink, and Barnaghi, 2017, s.13.2). The discussion touches on the need for extending the ontology of GeoSPARQL and other standards to arrive at a comprehensive spatial ontology that would better support multiple representation, coordinate reference systems, and links to non-spatial concepts. No specific method is proposed however, and for the time being, any shortcomings to current standards are left to each publisher to solve.

2.3. Related studies

Ordnance Survey (OS), the UK's national mapping agency, has published four of its data products as Linked Data: *Boundary-Line* (administrative units), the *1:50 000 Scale Gazetteer*, *Code-Point Open* (postcode units), and *Open Names* (OS, 2017). Goodwin, Dolbear, and Hart (2008) describe the creation of the first product. A hierarchy of boundary levels was defined, along with a set of topological relations to represent containment and adjacency. To encode the data according to these rules, they designed a purpose-built vocabulary with RDFS. It only described relative positions of features to each other, and no coordinate information was included. The authors also discuss URI strategy, data discovery, and linking to external data.

Vilches et al. (2010a; 2010b; 2014) studied methodologies for publishing data from several Spanish government agencies, covering hydrographical toponymy, administrative and statistical units, meteorological data, as well as social and economic statistics. After harmonization of the differently structured data sets, they were transformed and stored separately. Links between corresponding entities within them were made afterwards. Vocabularies were both developed in-house and adapted from other sources, including the Basic Geo Vocabulary. Of special interest is that these studies involved data sets at multiple scales.

The United States Geological Survey (USGS) initiated an effort to transform data sets from *The National Map* into RDF (Bulen, Carter, and Varanka, 2011; Usery and Varanka, 2012). It involved data conversion and ontology building (in OWL) for eight themes (land cover, structures, boundaries, hydrography, geographical names, transportation, elevation, and orthoimagery) covering nine test areas. The original data sets were converted to GML, from which geometries, attributes, and spatial relations were extracted and written in N3 format.

Shvaiko et al. (2012) describe an effort to transform a large portion of the geodata maintained by the provincial government of Trentino in Italy. The data were first converted to XML and then transformed into RDF. The geometries were encoded using a combination of the Basic Geo Vocabulary and a purpose-built extension. The authors place importance on the inclusion of metadata, which was encoded using shared vocabularies only. They also describe their methods for linking the features to external data and making the products available for use.

Consoli et al. (2014; 2017) similarly performed a transformation of geodata maintained by Catania municipality in Italy, mostly relating to infrastructure and public services themes. The original data were processed by a software tool that generated an executable script, which was then run to produce the output triple data and an accompanying vocabulary, both in Turtle format. At first, they used NeoGeo for encoding geometries, but later moved towards GeoSPARQL.

Atemezing et al. (2014) investigated the feasibility of publishing French national data sets with available vocabularies. As a case study, they transformed two data sets from the French national mapping agency (IGN), namely administrative units and toponyms, and linked them to external data. This was done by extending GeoSPARQL to describe the feature types and their geometries. As an alternative to including CRS information within geometry literals, they also created a custom vocabulary for CRSs, where each geometry object has a CRS object.

Patroumpas et al. (2015) discuss strategies for publishing and discovery of INSPIRE-compliant data and metadata on the Semantic Web. As a case study, they transformed seven Greek authoritative data sets relating to the INSPIRE themes geographical names, administrative units, addresses, cadastral parcels, transport networks, hydrography, and protected sites. After aligning the original data with INSPIRE schemas, each theme was transformed from GML to RDF/XML. The GeoSPARQL standard was utilized throughout, with geometries represented as WKT.

Hietanen, Lehto, and Latvala (2016) explored the feasibility of an on-the-fly transformation to RDF from an existing source. As a case study, they created an OWL vocabulary around an existing UML/GML data model used for Finnish geographical names, adding GeoSPARQL for describing spatial objects. Next, they decided on a URI strategy based on a national recommendation and proceeded to a prototype implementation. The approach taken was to set up a service that retrieves data through a WFS request from an existing system. The GML response is then transformed to RDF on-the-fly and returned in a requested output format.

Kadaster, the Dutch national cadastral and mapping agency, has published eight data sets as Linked Data on its PDOK Data Platform (Kadaster, 2016), the largest ones being the Key Registers Addresses and Buildings (*BAG*), Key Register Cadastre (*BRK*), Key Register Topography (*BRT – TOP10NL*), and the National Roads Registry (*NWB*). Detailed building geometries are found in both *BAG* and *BRT*, but there do not appear to be explicit links between corresponding entities across data sets. Beek et al. (2017a; 2017b) list the numerous novel functionalities of the platform's user interface, which include accessible schema- and instance navigation options, a query editor with highlighting, suggestions, and syntax checking, as well as a map view of query results. The platform is built around GeoSPARQL and WKT.

Debruyne et al. (2017) describe their ontology- and URI strategies for transforming administrative- and census boundaries data from Ordnance Survey Ireland (OSi). GeoSPARQL and a custom extension of it was used, with geometries (represented as WKT) acting as attributes of their respective features. Of special interest is that geometric representations at different LoDs for different purposes were produced, and each of those was then stored separately. They also investigated the ability to represent changes in boundaries over time.

Other studies have concerned transforming INSPIRE data from GML (its recommended format) to RDF, e.g. (Tschirner, Scherp, and Staab, 2011) and (Brink et al., 2014). Initially, this involved deriving OWL vocabularies from existing GML schemas to encode the data. Despite some success, it became apparent that an agreed upon methodology for creating vocabularies for the INSPIRE application schemas would be of benefit before pursuing the data transformation itself. This need led to preliminary work described by Tirry et al. (2015), the progress of which is introduced in Section 3.4.2.

3. Materials

3.1. Study area

The chosen study area is the city of Lund in Skåne County, Sweden, demarcated by a 6x6 km square (coloured red in Figure 4) covering most of the city's urban area. Its western and eastern boundaries are 13.144 668° and 13.242 590° longitude respectively, and its southern and northern boundaries 55.679 809° and 55.735 089° latitude (in WGS 84 decimal degrees). The densely built city centre lies just south of the middle, and a hospital area and a university campus with many large buildings stretch to the northeast. The bulk of the urban area consists of residential buildings of various types. Industrial and commercial buildings are found near the edges of the city, and farms occupy the corners of the study area.

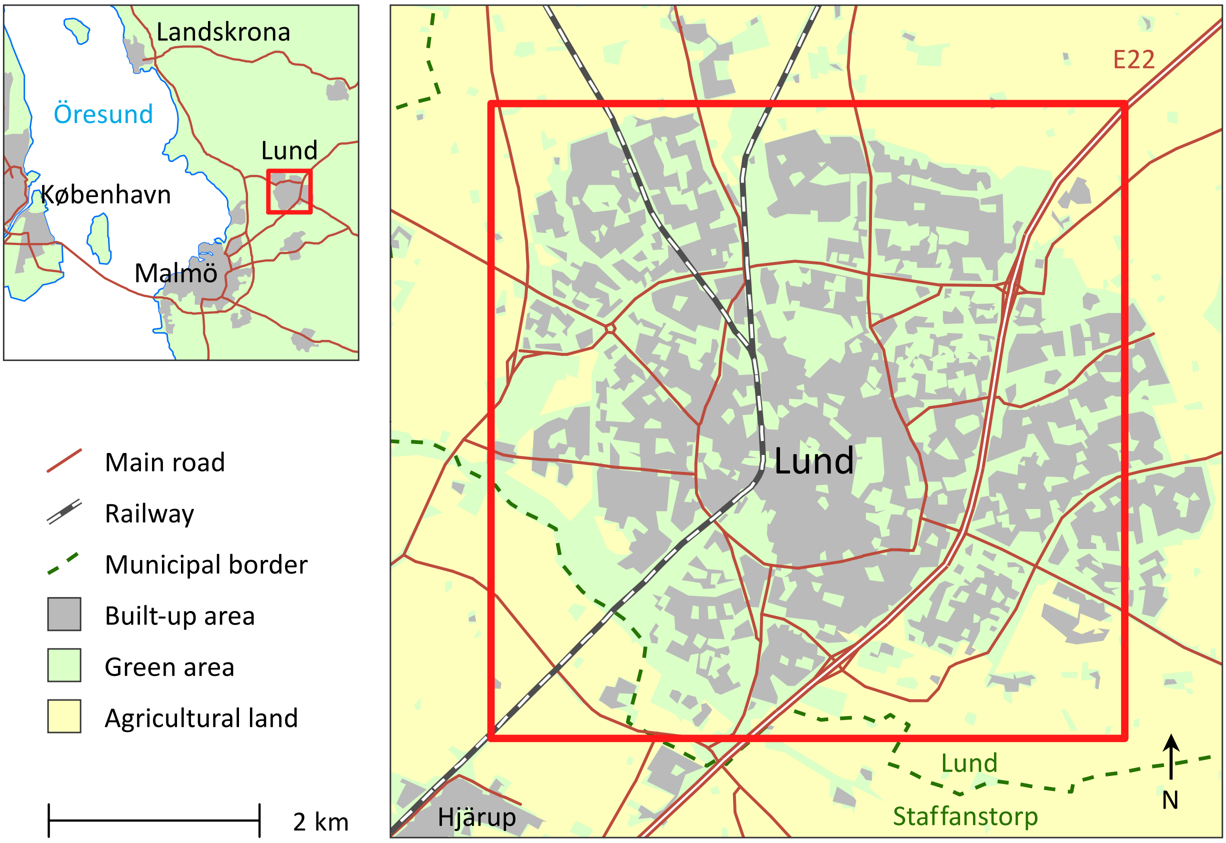


Figure 4. Extent of the study area (red). Data source: Lantmäteriet (2017b; 2017g; 2018).

3.2. Authoritative building data

Authoritative building data sets from three collections were acquired from *Lantmäteriet*, the Swedish mapping, cadastral and land registration agency. All three data sets are from the vector version of the respective collection, encoded in the SWEREF 99 TM coordinate reference system, and delivered in ESRI shapefile format. They are described below, ordered from the highest to the lowest level of detail.

3.2.1. Property Map

The GSD-Property Map (sv: *GSD-Fastighetskartan*) data collection (Lantmäteriet, 2017b), hereafter referred to as the Property Map or *fk*, holds the most detailed national data sets. It is divided into two parts: a cadastral part which contains 14 individual data sets; and a topographic part which contains 29 individual data sets within five thematic groups. These are documented in (Lantmäteriet, 2017c; 2017d). The stated reference scale for the whole collection is 1:10 000, and the stated scale range is 1:5000–1:20 000. At the time of writing, the data sets of the Property Map are not open data, although this could change in the future. A licence is needed for their use, so for the time being, they are not suitable for publication on the Web.

There are four data sets in the buildings/structures group, identified by names starting with "B". The high LoD polygon data set *BY* is of primary interest, and is described in detail below. *BA* is a polygon data set featuring various planar facilities such as sports fields, camping sites, cemeteries, or airfields, to name a few. *BO* is a line data set mostly concerning coastal facilities such as piers or quays. *BS* is a point data set for vertical structures such as lighthouses, wind turbines, or masts, but may also include churches. Features that exceed an area of about 15 m^2 may also be represented in *BY*.

The *BY* data set contains the most detailed building polygon representations, as can be seen in Figure 5a. It is based on a national database maintained by Lantmäteriet in collaboration with municipalities. All municipalities in the country participate in updating building data, either through regular delivery, taking place at least twice per year, or continuously through service-based updating. Municipalities collect data through geodetic surveying and various other means. This normally goes hand in hand with their other responsibilities, so updating is often concentrated in high-priority areas. Remaining areas are covered by Lantmäteriet's periodic collection of aerial photographs.

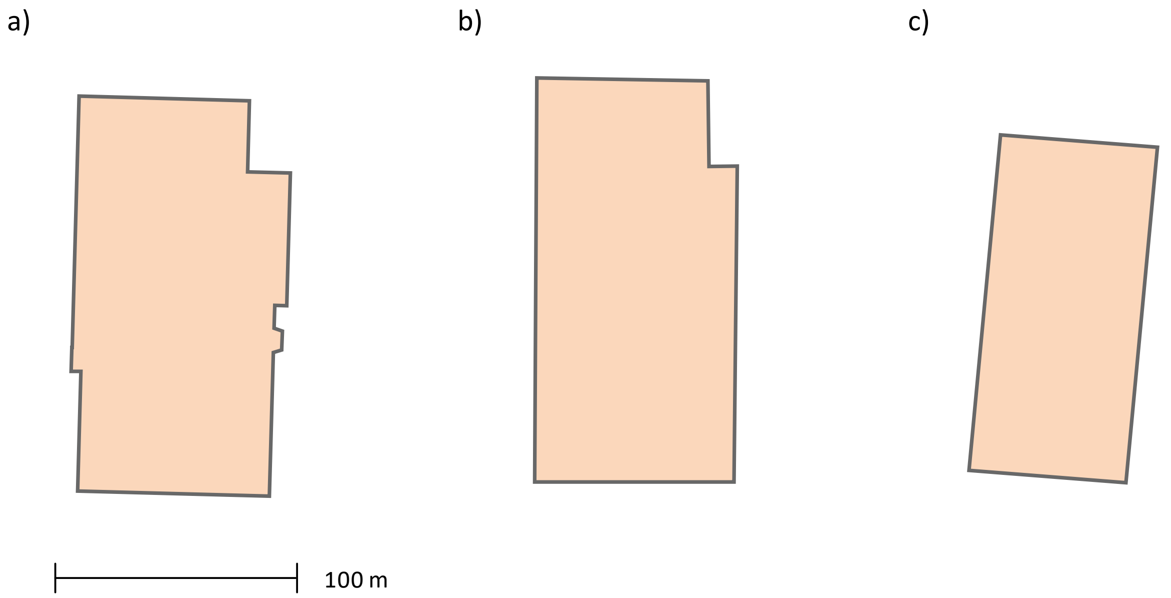


Figure 5. Example of a large building represented in the three authoritative data sets: a) GSD-Property Map, b) GSD-Topographic Map, and c) GSD-Road Map. Data source: Lantmäteriet (2017b; 2017g; 2017j).

The *BY* data contain 21 attribute fields, although only about half of those are typically filled in. *OBJEKT_ID* contains a unique identity code consisting of letters, digits, and hyphens, generated at the agency for each building feature. *OBJEKT_VER* and *ADAT* specify the version of the geometry and the date and time of last modification (at the time of last compilation). The way in which the geometry was measured, is specified by *INSAM_LAGE,* which can be one of four values: 0 (unspecified), 1 (facade), 3 (roof edge), or 4 (illustrative). *XYFEL* states the mean error in mm for the given method. The Boolean *HUVUDBYGGN* may be used to mark the main building within a cluster. Up to three building names may be given by *NAMN1*, *NAMN2*, and *NAMN3*. The remaining attribute fields are for categorizing building features. All features fall within one of three *DETALJTYP* categories: "HUS" (residential, government, business etc.), "HUSÖVR" (industrial and other), or "KYRKA" (religious). Further, each feature is labelled with one building purpose *ANDAMAL_1*, which is a three-digit code representing one of 49 categories, and a text version of the category *ANDAMAL_1T*. Additional building purposes may be given by *ANDAMAL_2* through *ANDAMAL_10* using the same codes.

More information on how the national database is maintained can be found in a handbook on buildings from Lantmäteriet (2017e). The agency's preferred way of measuring a building is along the "roof edge", which is usually measured photogrammetrically. It follows the highest envelope around the building and includes any appendages as long as they are regarded as parts of the same building object and lie inside the same property. The "facade" method is not well defined; it usually indicates collection by geodetic surveying (Lantmäteriet, 2017f) but it is unclear where the points are taken. A simple placeholder geometry for a building that has not yet been measured, e.g. a recent or planned construction project, is marked as "illustrative". Building polygons are allowed to overlap (Lantmäteriet, 2017e) which might be accurate e.g. when a balcony extends over another object.

3.2.2. Topographic Map

The GSD-Topographic Map (sv: *GSD-Terrängkartan*) data collection (Lantmäteriet, 2017g), hereafter referred to as the Topographic Map or *tk*, holds the second most detailed national data sets. It contains 38 individual data sets within 13 thematic groups, and is documented in (Lantmäteriet, 2017h; 2017i). The stated reference scale for the whole collection is 1:50 000, but an official scale range is not given. This applies to a typical rendering on a raster map with pre-defined symbolizations. However, the vector geometries are produced at a reference scale of 1:25 000, which can therefore be regarded as a lower bound. The data are licensed under Creative Commons CC0 (copyright waiver) and are thus readily available for publication on the Web.

There are three data sets in the buildings/structures group, identified by names starting with "B". They are largely collected by the same means as the Property Map, but have been generalized. *BL* is a line data set for various facilities such as recreational areas, piers, or airfields. *BS* is a point data set for representing 37 types of buildings, structures, and facilities with symbols. These include buildings of different sizes and functions, recreational facilities, masts, piers, cemeteries etc. *BY* is a polygon data set for large buildings, and includes runways as well. Compared to the Property Map, the geometries are simplified (Figure 5b) and are often aggregated. They are all single-part and disjoint from each other. The data contain only two attribute fields: *KKOD*, which is a three-digit feature category code, and *KATEGORI*, which is a text representation of the same code. *KKOD* can have one of two values: 690 (large building), and 728 (runway). The features are not marked with a unique identity code.

3.2.3. Road Map

The GSD-Road Map (sv: *GSD-Vägkartan*) data collection (Lantmäteriet, 2017j), hereafter referred to as the Road Map or *vk*, holds the third most detailed national data sets. It contains 31 data sets within eight thematic groups, and is documented in (Lantmäteriet, 2017k; 2017l). The stated reference scale for the whole collection is 1:100 000, but an official scale range is not given. However, the vector geometries are produced at a reference scale of 1:50 000, which can therefore be regarded as a lower bound. The data are licensed under Creative Commons CC0 (copyright waiver) and are therefore readily available for publication on the Web.

The buildings/structures group, contains the same three data sets as the Topographic Map: *BL*, *BS*, and *BY*. The line-based *BL* mostly includes bridges. The point-based *BS* symbol data set has much the same categories as in the Topographic Map, but adds a few sites of interest to drivers. The polygon data set *BY* contains buildings with very large footprints, usually located outside city centres, as well as runways. As can be seen in Figure 5c, the level of detail is low. The Road Map data contain the same two attribute fields as the Topographic Map: *KKOD* and *KATEGORI*. The possible values are the same, except the code 729 is used for runways. The features are not marked with a unique identity code.

3.3. Community gazetteer data – GeoNames

The community data chosen for data linking (Chapter 4) came from the GeoNames project, a hybrid authoritative/crowd-sourced gazetteer (toponym index). The project's data (GeoNames, 2017) are licensed under Creative Commons CC-BY and are available through direct download (global data set and country extracts) and through various web services (typically supporting XML and JSON as output). They are derived from a variety of sources, such as NMAs, municipalities, statistical agencies, travel sites etc. Among Swedish data providers are the National Heritage Board (RAA), Lantmäteriet, Statistics Sweden (SCB), and the Swedish Election Authority. Data can also be added or edited by volunteers via a provided user interface.

The country extracts consist of tab-delimited text files, with each line representing a GeoNames feature (gazetteer entry) in 19 fields. The first field, *geonameid*, is a unique numerical feature identifier. *name* is the default local place name and is followed by a plain *asciiname* version. Further place names (usually in other languages) can be provided in *alternatenames*. The location is stored as WGS 84 coordinates in the fields *latitude* and *longitude*. The field *feature_class* specifies one of nine top-level categories in the form of a single letter, e.g. "P" (settlements) or "S" (buildings/facilities), and *feature_code* specifies a more specific 2–5 letter category. Administrative units may be specified in *country_code*, *cc2*, *admin1_code*, *admin2_code*, *admin3_code*, and *admin4_code*. The remaining fields are *population*, *elevation*, *dem*, *timezone*, and *modification_date*.

An effort has been made to expose GeoNames to the Semantic Web. Each entry in the data has been given its own URI, based on the *geonameid* field. The project's *search* web service, which can return entries according to a number of parameters, supports RDF/XML as output. The RDF data are encoded using the project's purpose-built vocabulary (GeoNames, 2012), which includes a class *Feature*, properties for the fields (e.g. *name* and *alternateName*), some spatial relations (e.g. *parentADM1* and *nearbyFeatures*), as well as classes for the categories (e.g. *S.UNIV*). The point coordinates are encoded with the Basic Geo Vocabulary. The GeoNames vocabulary is not used directly during the data linking in Chapter 4 but appears in some queries in Chapter 5.

3.4. Shared vocabularies

3.4.1. GeoSPARQL

The GeoSPARQL standard (OGC, 2012a) is simultaneously a specification for a query language and a data model. Building on existing geospatial standards, it extends SPARQL with spatial functions, defines an RDF vocabulary for spatial objects, and provides RIF (Rule Interchange Format) rules for query transformation. Non-topological query functions (such as *geof:distance* and *geof:buffer*) are provided in addition to topological ones for three relation families: Simple Features, Egenhofer, and RCC8.

The main classes, properties, and data types of the GeoSPARQL vocabulary, hereafter known by the prefix *geo*, are shown in Figure 6. Two distinct classes *geo:Feature* and *geo:Geometry* are introduced, which are both subclasses of the abstract class *geo:SpatialObject*. These extensible top-level classes represent the separate concepts of a real-world object on one hand, and a geometric representation on the other. They are connected together by the property *geo:hasGeometry*, where the feature is the subject and the geometry is the object. A feature can have any number of geometries, and to mark one of them as the preferred geometry for visualization and spatial calculations, a second property *geo:hasDefaultGeometry* is provided, which is a sub-property of *geo:hasGeometry*. Unfortunately, this is erroneously realized as *geo:defaultGeometry* in the official implementation of the vocabulary (OGC, 2012b). A feature can technically have more than one default geometry, though it is not recommended.

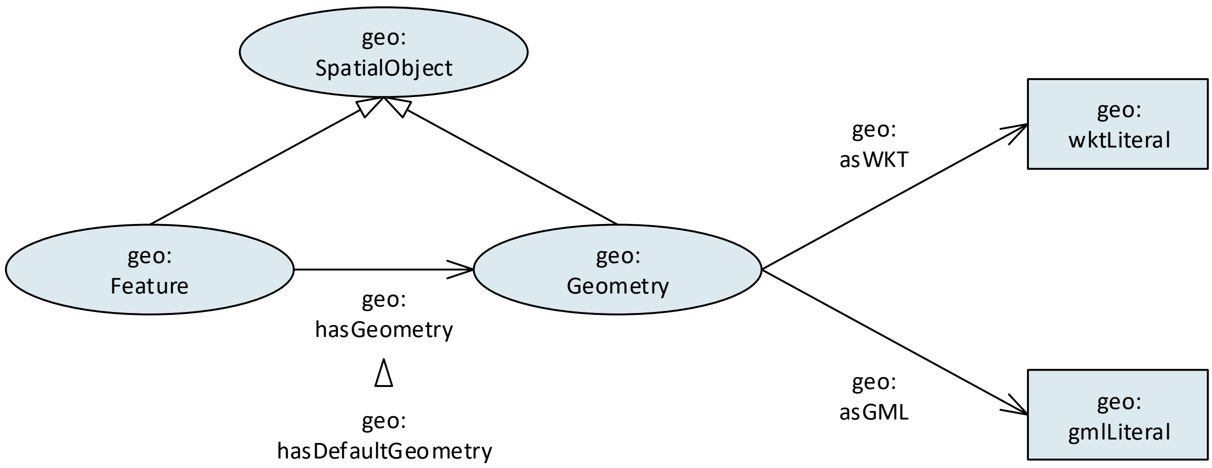


Figure 6. Classes and properties of the GeoSPARQL vocabulary as specified in (OGC, 2012a).

Currently, two geometry serializations are supported, i.e. formats for encoding the coordinate values as RDF literals. The first is WKT, for which a data type *geo:wktLiteral* is provided, and the other is GML, for which there is a data type *geo:gmlLiteral*. They are connected to *geo:Geometry* through the data type properties *geo:asWKT* and *geo:asGML*, which are both sub-properties of *geo:hasSerialization*. In the case of WKT, information on the CRS used may optionally be specified by adding a URI at the beginning of the string itself. The URIs for the CRSs are provided by the OGC. If no such URI is given, the WKT is assumed to be in WGS 84. In GML, this is done using the *srsName* attribute.

In conjunction with GeoSPARQL, OGC has released a supplementary Simple Features vocabulary (OGC, 2012b), hereafter known by the namespace *sf*, which follows the ISO/OGC data model of the same name (OGC, 2011). Therefore, a geometry can be encoded as an instance of a specific type like *sf:Point* or *sf:Polygon*. The *sf:Geometry* class at the top of the hierarchy is defined as a subclass of *geo:Geometry*, so an instance of any *sf* class can act as an object of *geo:hasGeometry* and have any of the supported serializations.

3.4.2. INSPIRE

INSPIRE (Directive 2007/2/EC) is the legal framework establishing a European Union spatial data infrastructure (SDI), and applies to data held by public authorities. While it is best known in relation to metadata and geoportals, there has also been work by thematic working groups to create common application schemas for the 34 spatial data themes within its scope (commonly referred to as INSPIRE themes). The thematic data models are maintained by the EC in UML and have been mapped to GML. Recently, focus has been put on mapping the UML models to RDF as well (Tirry et al., 2015). This ongoing work has resulted in draft guidelines (JRC, 2017a) on how RDF vocabularies for INSPIRE data themes should be constructed.

An initial set of draft INSPIRE vocabularies (JRC, 2017b) for selected themes has been published for review. The current version (0.2) includes seven INSPIRE themes: addresses, administrative units, buildings, cadastral parcels, geographical names, hydrography, and transport networks. In addition, it includes vocabularies for two generic data models that the others might depend on: base types and networks. The buildings theme is spread across three separate vocabularies, named *bu-base*, *bu-core2d*, and *bu-core3d*, which follows the three core profiles described in the data specification on buildings (EC, 2013; n.d. a). The *extended profiles* proposed in the specification are not supported.

Figure 7 shows the overall class structure of 2D buildings in the original schemas. Property names in the derived vocabularies usually follow the pattern <namespace>:<Class>.<property>. The superclass *bu-base:AbstractConstruction* has general attributes relevant to all types of buildings. The implemented properties are: *.contitionOfConstruction* (e.g. functional or ruin), *.dateOfConstruction*, *.dateOfDemolition*, *.dateOfRenovation*, *.elevation*, *.externalReference* (links to data sets providing additional information about an object), *.heightAboveGround*, and *.name*. *beginLifespanVersion* and *endLifespanVersion* hold the transaction time of a record version and are defined in *base*. *bu-base* does not define classes and properties needed for the *.name* attribute but reuses the *gn* vocabulary, which follows the data specification on geographical names (EC, 2014).

The class *bu-base:AbstractBuilding* inherits the properties of *bu-base:AbstractConstruction* and adds the following: *.buildingNature* (physical category), *.currentUse* (use category), *.numberOfDwellings*, *.numberOfBuildingUnits*, and *.numberOfFloorsAboveGround*. It has two subclasses: *bu-base:Building* and *bu-base:BuildingPart*, neither of which has any attributes. A building may consist of any number of parts through the property *bu-base:Building.parts*. The actual instances to be created belong to classes defined in *bu-core2d*. *bu-core2d:Building* and *bu-core2d:BuildingPart* inherit from *bu-base:Building* and *bu-base:BuildingPart* respectively, and each contains an attribute *bu-core2d:geometry2D* which is one or more instances of type *bu-base:BuildingGeometry2D* (see below).

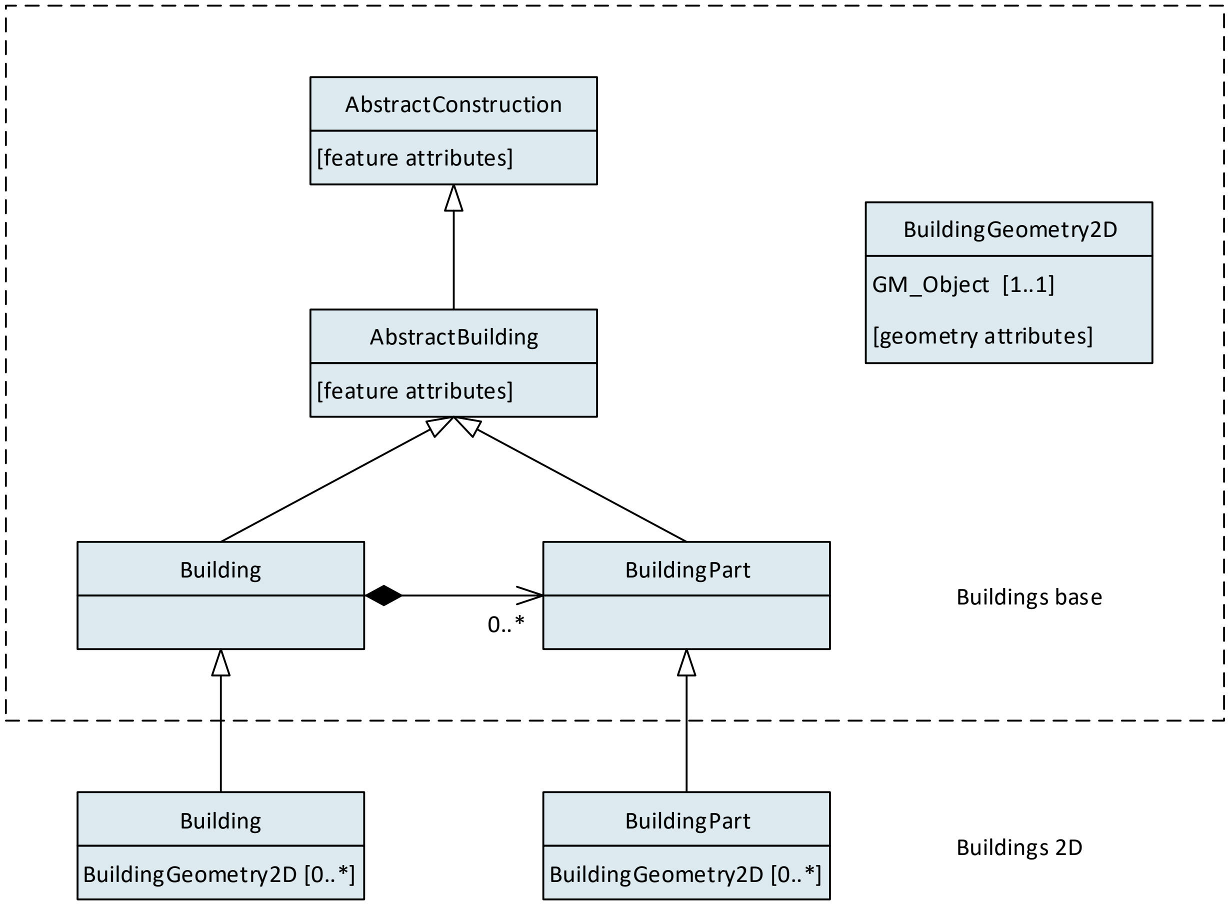


Figure 7. Simplified class diagram for 2D buildings in the INSPIRE data specification (EC, 2013; n.d. a).

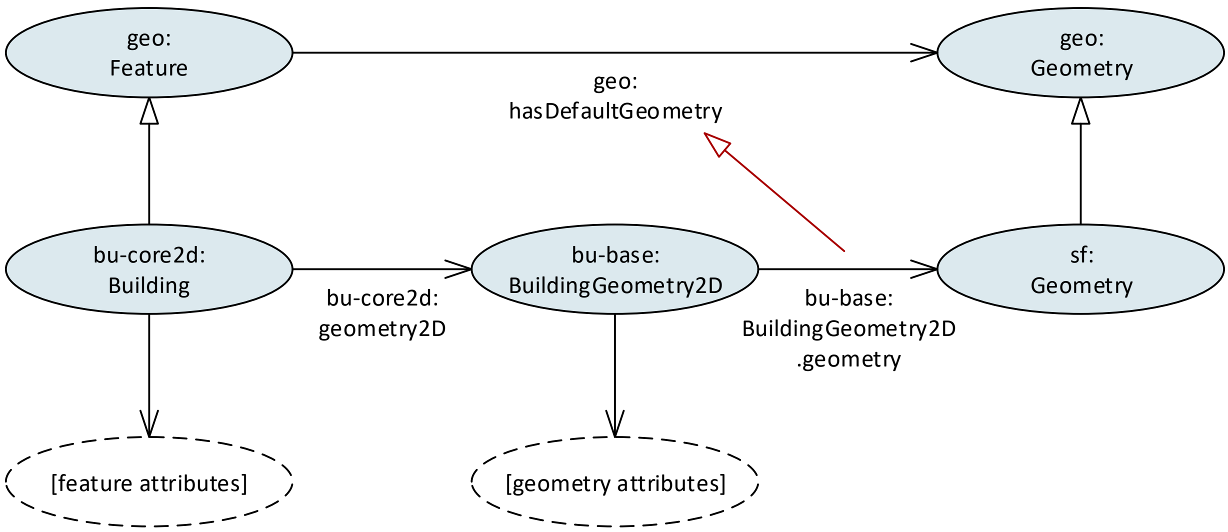


Figure 8. Simplified diagram of the relationships between the main classes as realized in Version 0.2 of the INSPIRE vocabularies (JRC, 2017b).

The INSPIRE vocabularies are integrated with GeoSPARQL. *bu-base:AbstractConstruction*, *bu-base:AbstractBuilding*, *bu-base:Building*, *bu-base:BuildingPart*, *bu-core2d:Building*, and *bu-core2d:BuildingPart* are all subclasses of *geo:Feature*. The data specification on buildings differs from most other themes in that instead of geometries being attributes of features, a special geometry class is defined, which is an attribute of a building feature and has a geometry object as an attribute. This is implemented with the class *bu-base:BuildingGeometry2D*, which is connected to *bu-core2d:Building* and *bu-core2d:BuildingPart* through the property *bu-core2d:geometry2D*, and has an attribute of type *sf:Geometry* through the property *bu-base:BuildingGeometry2D.geometry*. *bu-base:BuildingGeometry2D* has the additional attributes: *.referenceGeometry* (which marks it as the default geometry for viewing purposes), *.horizontalGeometryReference* and *.verticalGeometryReference* (the points of measurement), *.horizontalGeometryEstimatedAccuracy*, and *.verticalGeometryEstimatedAccuracy*.

Figure 8 shows the main classes and properties as they are used in practice. As can be seen, the *bu-base:BuildingGeometry2D* class is mapped to RDF in an unsound way, as it reads "feature has geometry" and then "geometry has geometry". Its property, *bu-base:BuildingGeometry2D .geometry*, is defined as a sub-property of *geo:hasDefaultGeometry*, which falsely implies that *bu-base:BuildingGeometry2D* is the same concept as *geo:Feature* (the domain of the super-property). It also means that a non-default geometry under *bu-base:BuildingGeometry2D .referenceGeometry* cannot be non-default in accordance with GeoSPARQL. The *.geometry* property can be used this way, but with possible conflicts. Alternatively, *bu-core2d:Building* (a subclass of *geo:Feature*) can be linked directly to *sf:Geometry* through *geo:hasGeometry*, but this would make the geometry attributes inaccessible.

3.4.3. SKOS

The Simple Knowledge Organization System (W3C, 2009) is a vocabulary that can be used to describe and interlink concepts. In addition to introducing the class *skos:Concept*, it provides classes and properties belonging to seven groups: concept schemes, lexical labels, notations, documentation properties (e.g. *skos:definition* and *skos:example*), semantic relations, concept collections, and mapping properties. The mapping properties group consists of five properties (plus a super-property *skos:mappingRelation*) for expressing alignment across different data models. *skos:broadMatch* and *skos:narrowMatch* are used for hierarchical alignments. Similar concepts can be associated with one another using *skos:relatedMatch, skos:closeMatch,* and *skos:exactMatch*, which are all symmetric, meaning that if A and B are related by a given property then B and A are also related by the same property.

4. Methods

4.1. Workflow

An overview of the tasks performed in the case study is shown in Figure 9. Data were read, transformed, and linked using Python 2.7, with geoprocessing relying on ArcGIS 10.5, primarily through ArcPy. The three authoritative building data sets described in Section 3.2 underwent initial preparation (Section 4.3) before being read by the main transformation process (Section 4.4). This transformation process handled the creation of features, geometries at three LoDs, attributes for feature objects, and attributes for geometry objects (which included scale information). The Property Map data (*fk_by*), being the most detailed of the three, served as a core data set from which both features and one representation level were derived. The other two provided additional representations (when available) which were matched geometrically with the most detailed geometries but linked to the corresponding features. Triples for class instances and links between them were constructed as Python strings, which were then written to an output Turtle document.

Next, gazetteer data from one RDF-enabled community project, GeoNames, were linked to the authoritative building data (Section 4.5). The GeoNames entries were matched with the most detailed building data set and linked to the previously derived feature objects. These links were written to a separate Turtle document. The two Turtle documents were then uploaded to a triplestore for storage (Section 4.6) along with relevant RDF vocabularies. This included a custom scale vocabulary (Section 4.2) designed to annotate geometries with explicit scale information (not considering LoD or resolution in general). Finally, a simple viewer was created to retrieve the stored data and visualize them in a web browser (Section 4.7).

Only polygon data (*BY*) were used from the authoritative collections. The collections also include point data (*BS*), but those are meant for producing map symbols and were not used in the transformation. The Topographic- (*tk_by*) and Road Map (*vk_by*) polygon data only represent selected buildings (those with large footprints). No attempt was made to generate new polygons to fill the gaps in these data sets. Records in the Property Map data (*fk_by*) containing temporary, or “illustrative”, geometries were excluded from the transformation. The related concept of built-up areas was not considered for use in the case study, as it describes land use rather than building features.

Since current vocabularies have only rudimentary ways of dealing with CRSs, and due to the open and interlinked nature of the Semantic Web, publishing geodata therein has so far almost universally been done using unprojected WGS 84 coordinates. There is a precedent for publishing geodata in RDF using a national system, e.g. (OS, 2017), and this remains a valid option for the Swedish data. It is also possible to store two separate geometries in different CRSs and create an application that requests the preferred one. However, for the sake of simplicity and easy integration with external Linked Data projects, a decision was made to follow the norm of using WGS 84 coordinates exclusively.

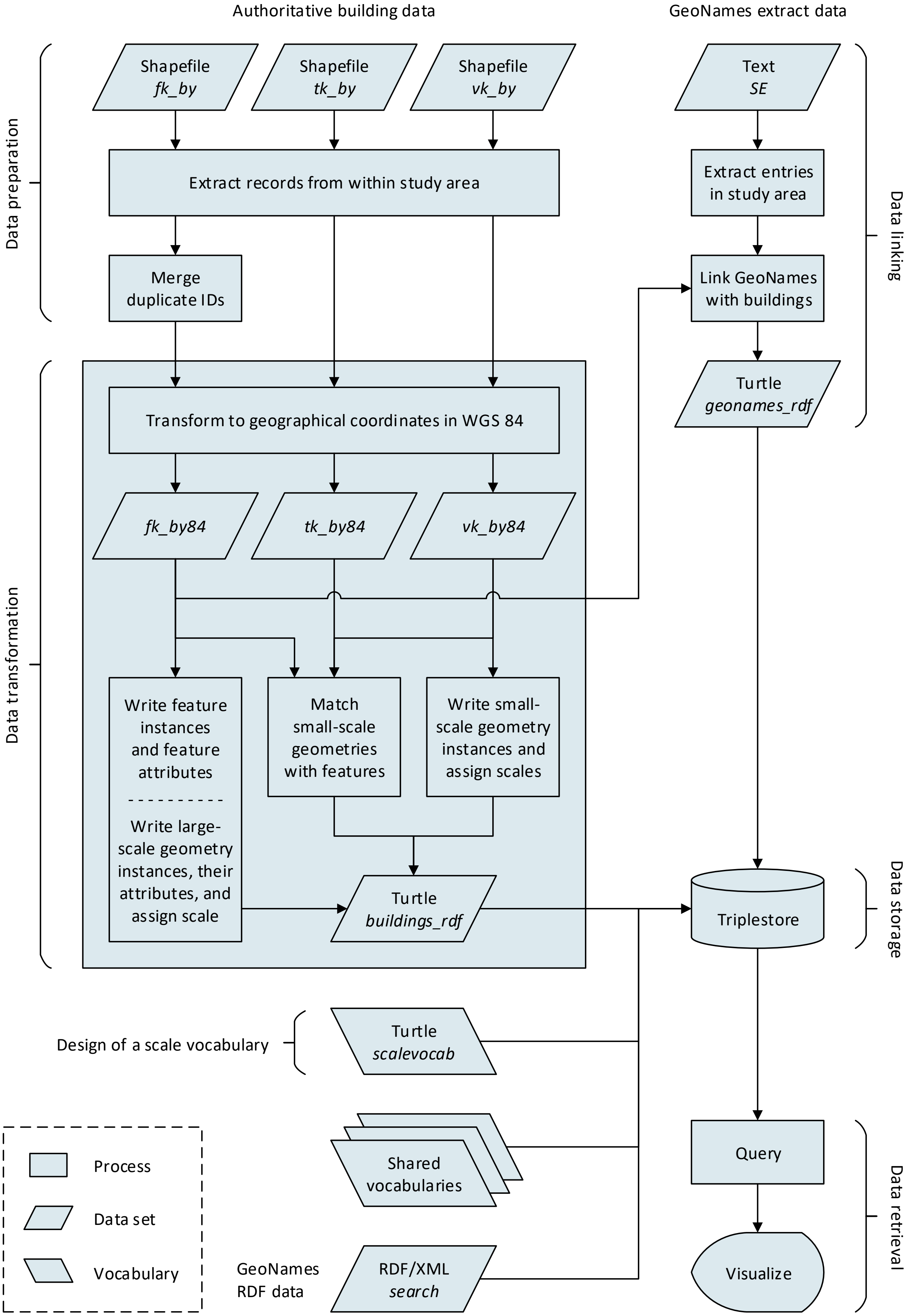


Figure 9. Overview of the workflow executed in the case study.

4.2. Design of a scale vocabulary

Due to the lack of a shared vocabulary for representing cartographic scale, a study-specific solution had to be devised. There are several options for designing such a vocabulary. One could be creating a number of custom sub-properties of *geo:hasGeometry*, as has been mentioned in Section 2.2.2, each with a scale level in its name. Another could be creating a number of custom subclasses of *geo:Geometry*, each with a scale level in its name. The disadvantage of these two options, apart from potentially requiring a vast number of class- and property definitions, is that the scale levels are bound to human-readable labels, and the objects encoded with them therefore cannot be discovered or selected through numerical filtering. A third option is to define a generic scale class for creating any number of scale instances that are then used to annotate geometry objects with numerical scale levels. This one was chosen for implementation in the study, and developed in conjunction with Huang et al. (2018).

Since the current INSPIRE buildings vocabularies effectively divide the concept of geometry into two classes, a decision had to be made on whether the scale information would be an attribute of *geo:Geometry* or the class *bu-base:BuildingGeometry2D* (see Section 3.4.2) which is used to store similar attribute data. Despite the appropriateness and universality of the former, the latter approach was chosen for implementation. This way, all geometry attributes (i.e. those describing geometry objects but not features) would be linked to the same class. By not stating a domain for the property however, the vocabulary was not restricted to this particular choice, and it can therefore be used to annotate a geometry of any theme or application schema. It would also be possible to assign a scale object to a whole data set rather than individual geometries, but this option was not explored in the case study.

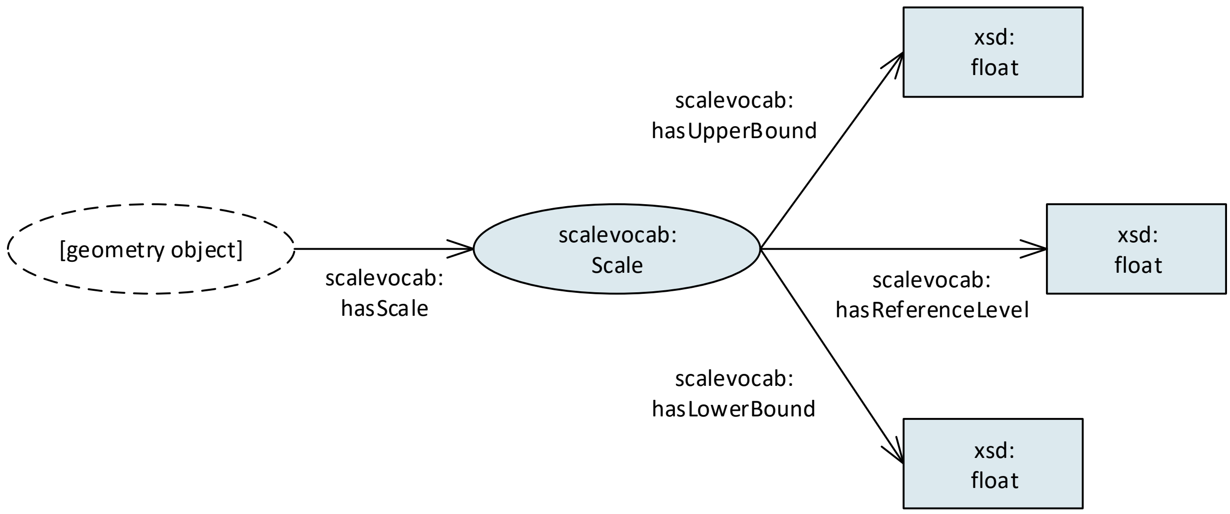


Figure 10. Diagram of the chosen design for the scale vocabulary.

The structure of the designed vocabulary can be seen in Figure 10. A custom object property, *scalevocab:hasScale*, was defined to associate a geometry instance with a scale instance. For the latter, a new class called *scalevocab:Scale* was defined along with three data type properties. The first, *scalevocab:hasReferenceLevel*, is meant for storing a data set's stated scale level, as provided by its producer. This would normally be the approximate scale at which the data are collected, the scale at which they are intended to be printed, or any other documented scale level. The other two properties, *scalevocab:hasUpperBound* and *scalevocab:hasLowerBound*, are for describing a data set's recommended viewing range, and are influenced by Carral et al. (2013). It should be noted that "upper" and "lower" refer to the numerical values, that is, the larger scale in a range becomes the lower bound and vice versa. Each of the three data type properties takes a denominator value in the form of a floating-point number, e.g. 1:10 000 is written as "10000.0". The full implementation of the scale vocabulary is included in Appendix A.

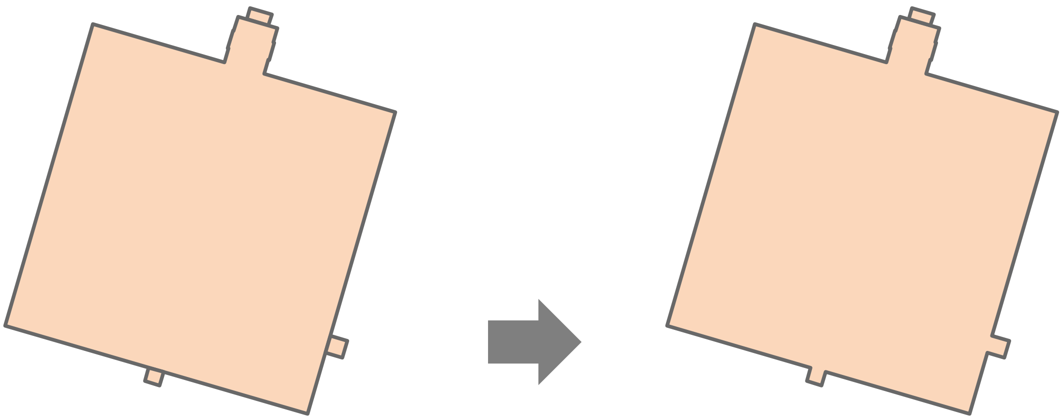
4.3. Data preparation

To represent the extent of the study area (Section 3.1), a 6x6 km square was created as a shapefile in SWEREF 99 TM, with a west–south–east–north bounding box: 383 500, 6 172 000, 389 500, 6 178 000. The three authoritative data sets are delivered as shapefiles in SWEREF 99 TM. For each of the three shapefiles, all records whose geometries fell within the square were extracted. This way, only whole polygons were included, with no cuts along the border. The extracts were named *fk_by.shp*, *tk_by.shp*, and *vk_by.shp* for the Property- Topographic- and Road Map respectively.

The unique identity codes of the Property Map (*fk_by*), stored in the *OBJEKT_ID* column (Section 3.2.1), provided an ideal source of identifiers for the construction of feature URIs. However, examination of the Property Map data revealed multiple duplicates among the records in this column. While the majority of records follow the norm of one ID per feature, some adjacent building parts and separate building objects on the same property shared the same ID. Since this would cause conflicts in the output data, and only the data producer can assign new IDs to features, the only available solution was to merge the records in question. This means that those duplicates whose geometries were disjoint resulted in the creation of multipart polygons, which were absent in the original data but should not cause problems for the subsequent RDF transformation.

A script was written for performing the task of merging records sharing the same IDs, relying on the Dissolve tool in ArcGIS. The full script is included in Appendix B. To begin with, a temporary ESRI geodatabase was created (line 10) since it supports more geoprocessing options than shapefiles do. Next, the records of *fk_by.shp* were dissolved based on the *OBJEKT_ID* column (line 18), while also keeping all other columns in place (*keepColumns*). The *multi_part* option was left as true because otherwise disjoint duplicates would not be merged. When records get merged into one feature, only one row of attribute values is preserved. Since selecting a representative record from the old ones is not a straightforward task, the option *statistics_fields* was set to "FIRST" for all columns, meaning that the values of the first occurrence among duplicates was kept. Although the values are often identical for a single feature, this did make it possible for some values to become redundant, namely in the columns *OBJEKT_VER*, *ADAT*, *INSAM_LAGE*, and *XYFEL*. These fields, if merged, were therefore marked (lines 13–17) and then set to null (lines 28–36).

The output, now stored in the geodatabase, was then saved as a shapefile (lines 39–41) overwriting the original *fk_by.shp*. This way, the data transformation step could commence as if the data set had not contained any duplicates in the first place. The process brought the number of records in *fk_by* from 24 654 down to 22 927. Figure 11 shows an example of a building whose records got merged during this process. The core part of the building (marked with an asterisk) has three small appendages, of which one (the front entrance) has its own ID but the other two share theirs with the core part. The Dissolve tool keeps only one duplicate row (not necessarily the core part) and replaces the relevant geometries with a merged one.



	OBJEKT_ID
	f1becfc7-cffc-44cc-a70c-8b93bb4de73f
*	450672a5-a274-494e-ba6a-4cf06cabb610
	450672a5-a274-494e-ba6a-4cf06cabb610
	450672a5-a274-494e-ba6a-4cf06cabb610

OBJEKT_ID
f1becfc7-cffc-44cc-a70c-8b93bb4de73f
450672a5-a274-494e-ba6a-4cf06cabb610

Figure 11. Example of merged duplicate IDs in the Property Map.

4.4. Data transformation

This section describes the transformation of the three authoritative data sets, here called *fk_by*, *tk_by*, and *vk_by*, into RDF form. A single script was created to perform all the tasks grouped under "data transformation" in Figure 9. The full script is included in Appendix C.

4.4.1. Initial steps

In order to allow the use of all processing tools in ArcPy, the three data sets were handled inside an ESRI file geodatabase *temp.gdb*, arranged in lines 10–11 (the one created in the previous section was in fact reused). The first task was to perform the transformation to the target CRS. Each shapefile, *fk_by.shp* (now without duplicate IDs), *tk_by.shp*, and *vk_by.shp*, were taken as input and transformed from SWEREF 99 TM to unprojected WGS 84 coordinates (lines 14–17). The output was stored as *temp.gdb/fk_by84*, *temp.gdb/tk_by84*, and *temp.gdb/vk_by84* respectively. These served as the input in subsequent steps (where they are simply referred to as *fk_by*, *tk_by*, and *vk_by* as before).

Next, an output document named *buildings_rdf.ttl* was created (line 20; referred to as *ttl* in the code) for writing out triples in Turtle format. Prefixes for nine vocabularies (eight shared and one custom) and eleven for instance URIs (see the following sub-sections) were placed at the top of the document (lines 23–42), followed by scale instances, small-scale geometry instances, and finally feature- and large-scale geometry instances.

4.4.2. Scale instances

As mentioned in Section 3.2, the collections that contain the data sets are maintained and published with certain printing/viewing scales in mind. These stated scales were employed for annotating the building geometries in the study, although it is important to note that they apply to the entire collections (e.g. all data sets belonging to the Topographic Map) and may not be optimal for viewing the theme of interest (such as buildings). The Property Map as a whole has a reference level of 1:10 000 and a stated range of 1:5000 to 1:20 000. The Topographic Map has a reference level of 1:50 000 but no stated range. However, the larger 1:25 000 is given as a possible viewing level and was therefore regarded as a valid lower bound substitute. The solution for the upper bound was to make it the same as the reference level. Similarly, the Road Map has a reference level of 1:100 000 and a possible viewing level of 1:50 000, so the substitute range was set to those levels. These ranges left a gap between 1:20 000 and 1:25 000, which is inconvenient but not a serious problem for the case study.

One scale instance was created for each collection (lines 45–59) as part of a namespace </shared/scale#>. Due to the very small number of instances, and their association with specific data products, they were given short descriptive IDs: “fk_10k”, “tk_50k”, and “vk_100k”. In addition to a link to the product’s website, each of the three scales had three values associated with it:

```
scalevocab:Scale scalevocab:hasReferenceLevel xsd:float

scalevocab:Scale scalevocab:hasUpperBound xsd:float

scalevocab:Scale scalevocab:hasLowerBound xsd:float
```

4.4.3. Small-scale geometry instances

Since no unique IDs are provided with the building records in the *tk_by* and *vk_by*, these had to be generated for use in the study. A new column *GEOM_ID* was added to the attribute table of each of them, which was then populated with a unique row number (lines 62–76). The IDs started at 1 000 001, solely for the purpose of making them easy to identify during testing.

Two namespaces were created for each representation level: </bu/polygonGeom50k#> for instances of *bu-base:BuildingGeometry2D*, and </bu/polygonGeomObj50k#> for instances of *sf:Geometry* (“50k” for *tk_by*; “100k” for *vk_by*). Each data set was looped through (lines 79–99) and its geometries were written to the output Turtle document using the previously created IDs. After creating the two instances, they were connected together along with the WKT with the two-triple pattern:

```
bu-base:BuildingGeometry2D bu-base:BuildingGeometry2D.geometry sf:Geometry

sf:Geometry geo:asWKT geo:wktLiteral
```

Next, a scale instance was assigned to the *bu-base:BuildingGeometry2D* instance:

```
bu-base:BuildingGeometry2D scalevocab:hasScale scalevocab:Scale
```

In addition to the scale, one INSPIRE geometry attribute was assigned to the same instance. This attribute specifies whether it is the default geometry, and was set to “false” for all occurring records in both *tk_by* and *vk_by*. The triple was in the form:

```
bu-base:BuildingGeometry2D bu-base:BuildingGeometry2D.referenceGeometry
    xsd:boolean
```

Figure 12 shows how the geometry attributes were linked to the small-scale geometries. It also applies to the large-scale geometries (next section).

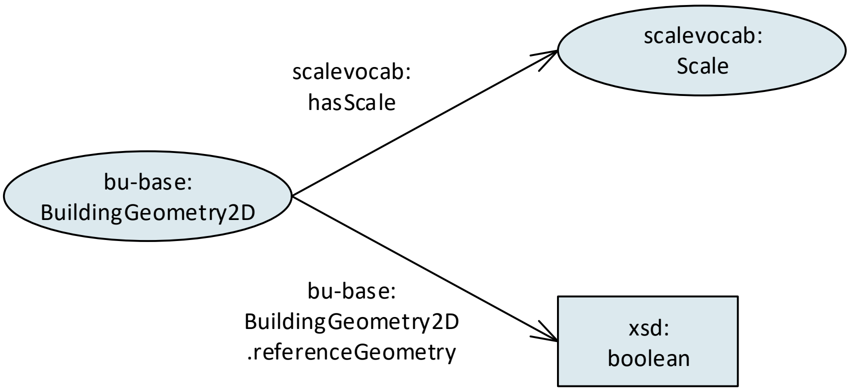


Figure 12. Diagram showing how geometry attributes were linked with geometry instances (both large-scale and small-scale geometries).

4.4.4. Feature- and large-scale geometry instances

The remainder of the transformation process (Sections 4.4.4–4.4.6) was performed in one large loop (lines 102–189) through the *fk_by* data set. This data set was used as a source for both feature instances and one representation level, meaning that the existence of a feature was determined by its occurrence as a record therein (with its own unique ID provided by the producer). Records containing placeholder geometries (marked as "illustrative" in the *INSAM_LAGE* column) were excluded entirely (line 104) since their representation level could not be easily determined.

A namespace </bu/feature#> was created to hold instances of *bu-core2d:Building*, and two more, </bu/polygonGeom10k#> and </bu/polygonGeomObj10k#>, to hold instances of *bu-base:BuildingGeometry2D* and *sf:Geometry* respectively. For each record in *fk_by*, a feature instance was created (lines 106–107) with an ID derived unchanged from the *OBJEKT_ID* column. The corresponding large-scale geometry was then added (lines 110–115) in the same way as in the previous section:

```
bu-base:BuildingGeometry2D bu-base:BuildingGeometry2D.geometry sf:Geometry
```

```
sf:Geometry geo:asWKT geo:wktLiteral
```

The 1:10 000 scale instance from Section 4.4.2 was then assigned to it (line 116):

```
bu-base:BuildingGeometry2D scalevocab:hasScale scalevocab:Scale
```

Next, the geometry had to be linked to its feature, since they were now separate objects. Due to the misalignment between geometry classes across the vocabularies (end of Section 3.4.2), this could be done in two separate ways (which should ideally work in tandem). The first would be to connect *bu-core2d:Building* with *bu-base:BuildingGeometry2D*, in accordance with the current INSPIRE vocabularies, and the other would be to connect *bu-core2d:Building* (which inherits from *geo:Feature*) straight to *sf:Geometry* using the GeoSPARQL vocabulary (using the official but erroneous *geo:defaultGeometry*; see Section 3.4.1). A decision was made to include both ways to ensure full conformance, and two triples were therefore created (lines 117–118):

```
bu-core2d:Building bu-core2d:geometry2D bu-base:BuildingGeometry2D
```

```
bu-core2d:Building geo:defaultGeometry sf:Geometry
```

In addition to the scale instance above, three geometry attributes from the INSPIRE schema were assigned to *bu-base:BuildingGeometry2D* (lines 121–125). The first specifies whether it is the default geometry, and was set to “true” for all instances at this representation level. The horizontal geometry reference specifies which part of the building was measured, and was derived from the *INSAM_LAGE* column. Since the meaning of the value “facade” is unclear (Section 3.2.1), this attribute was only derived from records with the value “roof edge”. Judging from Lantmäteriet’s handbook on buildings (Lantmäteriet, 2017e), the “roof edge” method corresponds to “above ground envelope” in (EC, 2013), so this value was used in all cases (rather than the narrower “roof edge” in the same specification). The resource URI for it is provided by the EC (n.d. b). The geometry’s estimated accuracy was derived from the *XYFEL* column, and was encoded using the prescribed *base* part of the INSPIRE vocabularies (here called *basetypes* to avoid errors in Turtle validators). The three geometry attribute triples were written as:

```
bu-base:BuildingGeometry2D bu-base:BuildingGeometry2D.referenceGeometry
    xsd:boolean

bu-base:BuildingGeometry2D bu-base:horizontalGeometryReference skos:Concept

bu-base:BuildingGeometry2D bu-base:horizontalGeometryEstimatedAccuracy
    basetypes:Measure
```

Due to the omission of “facade”, it was possible for a geometry instance to have an accuracy value but no horizontal reference value. Furthermore, no triples were written for these two last attributes if the record had been merged in Section 4.3. The scale instance and the *.referenceGeometry* attribute are common with the small-scale geometries (Figure 12). Attributes that were only assigned to large-scale geometries are depicted in Figure 13.

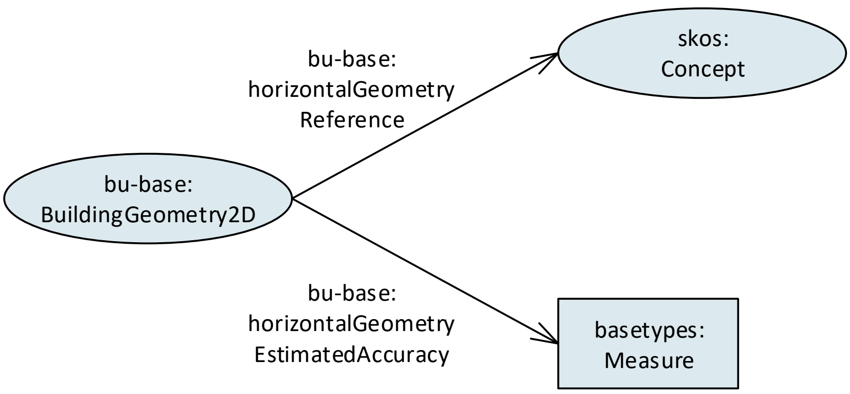


Figure 13. Diagram showing additional geometry attributes for large-scale geometries.

4.4.5. Matching small-scale geometries with features

Since each feature was tied to exactly one large-scale geometry (both coming from the same record in *fk_by*), the small-scale geometries could be matched to it geometrically. Two polygons can be matched together by various means, including advanced ones like in (Zhang et al., 2014), but investigating different matching techniques is outside the scope of this thesis. To accomplish the task, a simple overlap check was employed. Because a portion of one building’s representation often overlaps a representation of a nearby building, a threshold of some sort had to be applied in order to prevent false matches. It was decided that a feature and a small-scale geometry would match if at least half of the large-scale geometry fell inside the small-scale geometry. It is important to note that this method has limitations due to e.g. displacement of representations or unsynchronized updating across data sets, leading to some difficult cases.

Within the larger loop through *fk_by* (see the previous sub-section), two inner loops (lines 128–146) searched for matching building geometries in *tk_by* and *vk_by*. For each record in each of the two data sets, if its geometry was not disjoint with the geometry from *fk_by* (lines 131 and 141), it was regarded as a valid representation of the feature (lines 132 and 142) if the following criterion was met:

$$A_{S \cap L} \geq \frac{A_L}{2}$$

where *A* is the geometric area of a polygon, *S* is the small-scale geometry (of either *tk_by* or *vk_by*), and *L* is the large-scale geometry (of *fk_by*). As with geometries from *fk_by*, matched geometries from *tk_by* and *vk_by* were connected to corresponding features in two fashions (lines 135–136 and 145–146), following both the current INSPIRE vocabularies and the GeoSPARQL vocabulary:

```
bu-core2d:Building bu-core2d:geometry2D bu-base:BuildingGeometry2D
```

```
bu-core2d:Building geo:hasGeometry sf:Geometry
```

No attempt was made to force one-to-one relationships between representation levels; one small-scale representation could thus represent several features and be related to several large-scale representations. Additionally, due to the small-scale geometry instances being created separately (before the loop through *fk_by*), it was possible, though unlikely, that a geometry was created but not linked to any feature.

4.4.6. Feature attributes

Attributes in *fk_by* that describe the real-world entities were linked to the feature objects when available. Only those columns that have a corresponding attribute type in the INSPIRE schemas were included. Building names were assigned to features (lines 149–162) using a combination of the INSPIRE *bu* and *gn* vocabularies (Figure 14). All non-empty fields of the three building name columns (*NAMN1*, *NAMN2*, and *NAMN3*) from *fk_by* were included. A namespace </bu/buildingName#> was created to hold instances of *gn:GeographicalName*, and another </bu/buildingNameSpelling#> for instances of *gn:SpellingOfName*. To distinguish multiple names of the same feature, an ID was constructed for each instance using the feature ID (*OBJEKT_ID*) with an appended number from one to three (*nameNo*). Each attribute was added to the output file using the three-triple pattern:

```
bu-core2d:Building bu-base:AbstractConstruction.name gn:GeographicalName
```

```
gn:GeographicalName gn:GeographicalName.spelling gn:SpellingOfName
```

```
gn:SpellingOfName gn:SpellingOfName.text xsd:string
```

INSPIRE provides two attributes related to building type. The first is *buildingNature*, which refers to physical aspects of structures that may be of use in mapping. The other is *currentUse* which is covered by the *ANDAMAL* columns in *BY*. The less detailed *DETALJTYP* column was not used. *currentUse* provides five upper level categories, which are meant to bridge those used in most countries, and a total of eight lower ones (not counting possible extensions by the NMAs). For the sake of simplicity, only the upper level was used. The Swedish categories 120–199 (residential) were mapped to “residential”, 240–299 (industrial) to “industrial”, 301–399 (public) and 499 (commercial) to “commerceAndServices”, 599 (agricultural) to “agriculture”, and 699 (complementary) to “ancillary”. No category is provided for “other”, so no triples were created for category 799.

The steps of adding building use (Figure 14) were performed in lines 165–186. All ten building use columns from *fk_by* were evaluated. Only one link was created for each of the five upper categories, if that category was present for a feature, as ensured in line 166. In other words, a feature could be linked to several building uses, but the link to each category could only be made once per feature. A namespace </bu/buildingCurrentUse#> was created to hold instances of *bu-base:CurrentUse*. To distinguish multiple building uses of the same feature, as with building names, each instance ID reused the feature ID and appended a number (from one to three and five to six due to categories 3xx and 4xx being the same). The last part of the pattern has the range *skos:Concept* rather than strings, which takes predefined URIs provided by the EC (n.d. b), e.g. <http://inspire.ec.europa.eu/codelist/CurrentUseValue/residential> for residential buildings. Each resulting link was of the two-triple pattern:

```
bu-core2d:Building bu-base:AbstractBuilding.currentUse bu-base:CurrentUse
bu-base:CurrentUse bu-base:CurrentUse.currentUse skos:Concept
```

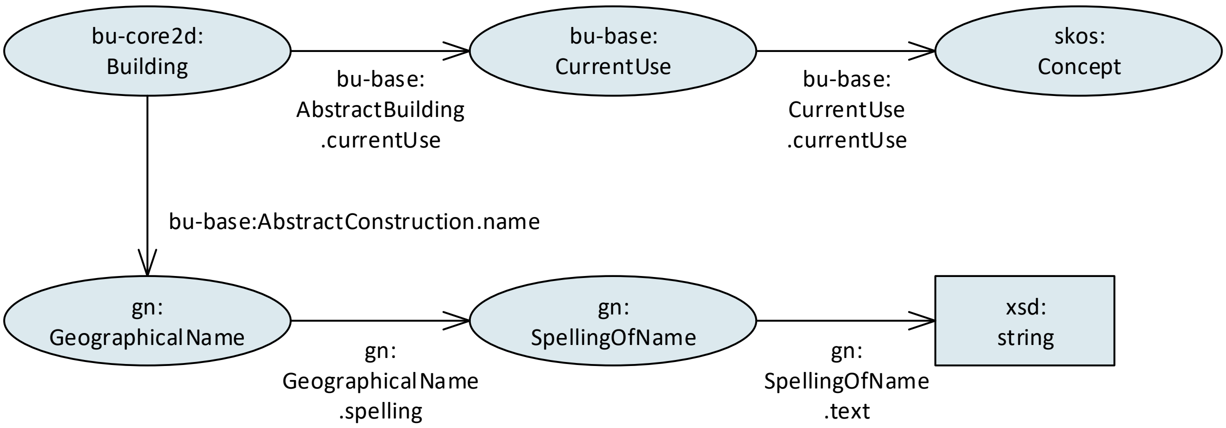


Figure 14. Diagram showing how feature attributes were linked with feature instances.

4.5. Data linking

A country extract of all the GeoNames data for Sweden, consisting of a tab-delimited text file *SE.txt*, was downloaded from the project's website (GeoNames, 2017). To reduce computing time, a short script (included in Appendix D) was created to extract entries that fell within the study area. The script evaluated the coordinates in fields 5–6 and wrote the rows of interest to a new file called *geonames_studyArea.txt*. This shorter (but otherwise containing the same information) file was used as an input in the subsequent linking.

A script was created for finding correspondence between building entries in GeoNames and features in *fk_by*. The script is included in Appendix E. A new document *geonames_rdf.ttl* was created (line 14) for writing the output triples in Turtle format. A loop through the records in *fk_by* (lines 35–49) first ensured that the geometry was not "illustrative" and that the building name columns were not empty (line 37), and then compared them with each row in the GeoNames data that had the top-level category "S" (buildings/amenities) in field 7 (line 43).

The two data sets could be matched by various means, such as by simple geometric overlap of *fk_by* polygons and GeoNames points, but instead, a method used by Stadler et al. (2012) to link GeoNames and OpenStreetMap data was adapted to the study. This method gives a ⅔ weight to a string similarity measure that compares the names themselves, and a ⅓ weight to a geometric measure between two points. These are then approved based on a given threshold (line 46). The former was performed (lines 44 and 21–25) on the default *name* value (field 2) from GeoNames and all three name fields from *fk_by* (*NAMN1*, *NAMN2*, and *NAMN3*), with only the best match taken into account. The Jaro-Winkler distance was calculated using an implementation by Ratte (2016). The alternate names in GeoNames were not considered. No attempt was made to exclude category names like “universitet” occurring within the names themselves. The geometric measure was performed (lines 45 and 27–32) using the WGS 84 point from GeoNames (fields 5–6) and the centre of gravity of the geometry from *fk_by*. The haversine distance was calculated using an implementation by Deniau (2016). The maximum allowed distance between objects was set to c = 100 m.

Once a match was found, a link had to be encoded with a suitable vocabulary. A popular choice is to use *owl:sameAs* for links between similar objects, but this is considered inappropriate in many cases (Goodwin, Dolbear, and Hart, 2008; Halpin et al., 2010). Since a gazetteer entry is not equivalent to a building feature, it was decided to use the predicate *skos:closeMatch* from the SKOS vocabulary (Section 3.4.3) for this purpose. This predicate is symmetric, so one triple acts as two (Figure 15). Each link was written to the output document as a single triple (line 47), using the *geonameid* field to replicate the GeoNames URI:

```
bu-core2d:Building skos:closeMatch geonames:Feature
```

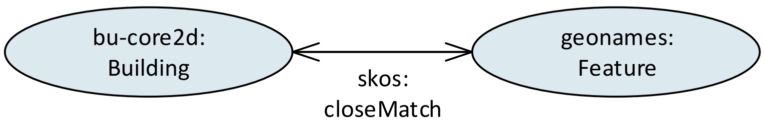


Figure 15. Diagram showing how GeoNames entries were linked with building features.

4.6. Data storage

The transformed and linked data, along with relevant vocabularies, were uploaded to a Stardog triplestore (graph database). The data came from two Turtle documents: *buildings_rdf.ttl*, created in Section 4.4, and *geonames_rdf.ttl*, created in Section 4.5. To be able to interact with the linked GeoNames data in their entirety, a copy of all relevant triples was downloaded through the project’s *search* function. Entries in category “S” falling within the study area were extracted by constructing the following request, and the output RDF/XML document was then uploaded to the triplestore:

```
<http://api.geonames.org/search?q=SE&featureClass=S&west=13.1446&east=13.2426&
    south=55.67980&north=55.73509&username=eidsson&type=rdf>
```

For the use case in Section 5.2.3, a copy of *CurrentUseValue* in RDF/XML format was obtained from the INSPIRE code list register (EC, n.d. b) and uploaded with the rest of the data. The uploaded vocabularies consisted of the custom scale vocabulary described in Section 4.2 and shown in Appendix A, and four parts of Version 0.2 of the INSPIRE vocabularies: *base*, *bu-base*, *bu-core2d*, and *gn* (JRC, 2017b). External vocabularies are not automatically imported into the triplestore, so copies of SKOS (W3C, 2009) and the GeoNames vocabulary (GeoNames, 2012) were uploaded as well. Stardog has native support for much of GeoSPARQL, but the separate

Simple Features vocabulary (OGC, 2012b) had to be added. Essentials like RDFS, OWL, and XML Schema data types were already present.

4.7. Data retrieval

For the purpose of querying and visualizing the stored data during the demonstrations in Chapter 5, a simple web viewer was created using Stardog.js and OpenLayers. It accepts a SPARQL query and sends it to the triplestore, which responds with a JSON output. Each value returned for a variable named *?wkt* in the query is then converted into an OpenLayers geometry and rendered in the browser using the commonly used Web Mercator projection. Additionally, the full JSON output is written to the browser's console for added insight.

The viewer was built with the option of automatically requesting representations at a given scale, read from the zoom level. Working with the representative fraction on digital devices is problematic. While a screen's pixel resolution is easily obtainable by the client, its physical dimensions are not. Therefore, any calculation of a digital map's scale is bound to be an approximation at best. OpenLayers provides functions for obtaining the map resolution from the set zoom level, given as projection units per pixel. Assuming a physical pixel width of 0.28 mm, the viewer calculates the scale for the set zoom level as metres per pixel divided by 0.000 28. It then replaces any occurrence of the string part "scaleFromZoom" in a query with the approximated scale level.

5. Results

5.1. Data output

The outcome of the data transformation and linking in Chapter 4 is briefly described here. The merging of duplicate IDs in Section 4.3 reduced the number of records in *fk_by* from 24 654 to 22 927 (including 17 with "illustrative" geometries that where ignored in a later step). The original data contained no multipart polygons, but the merging process introduced 41 of them. It should be mentioned that the eventual WKT literals were all written out as multipolygon objects. This is due to the way ArcPy returns polygon objects as WKT; all geometries (bar those 41) were in fact regular polygon objects. This issue did not complicate other aspects of the study in any way, so no attempt was made to modify the strings. It is, however, something to consider if ArcPy is to be used for working with WKT in future projects.

The transformation script created in Section 4.4 produced an output file *buildings_rdf.ttl*. Each line was populated with only one triple, without any predicate/object nesting. The syntax was validated during upload to the triple store, which reported no errors. The document began with a prefix header that included namespaces for both vocabularies and class instances. The header was followed by a hardcoded block for three scale objects (Section 4.4.2) with which geometries were later associated, consisting of 15 triples in total. The remainder of the file contained the feature- and geometry data, along with their attributes, totalling 316 218 triples. First came blocks of triples for 278 geometries from *tk_by* and nine from *vk_by* (six triples each). They were followed by blocks for 22 910 *fk_by* records (including triples for large-scale geometries), with 9–26 triples per block (the median being 14).

Although it is not the aim of the study to evaluate matching methods, it is worth looking at the effectiveness of those used. In Section 4.4.5, small-scale geometry instances were compared with large-scale ones using a simple geometric overlap check. Despite susceptibility to displacement, caused by the chosen 50 % threshold (half of a large-scale one inside a small-scale one), the number of difficult cases turned out to be minimal. Out of the nine geometries in the *vk_by* data set, only one failed to match with the main part of its *fk_by* equivalent. The corresponding geometry in *tk_by* did successfully match to the main part however.

In Section 4.5, a method combining geometric distance and string similarity was adopted to link GeoNames with the authoritative data (*fk_by*). Out of 31 GeoNames entries of category "S" in the study area, ten matches were returned and written to an output file *geonames_rdf.ttl* (one triple each). All ten matches had identical strings. Five potential matches of varying string similarity failed to match (including one multi-building complex that also missed the geometric maximum). Judging from this small sample, the threshold may be too strict. It is also worth noting that it was possible for one GeoNames name to match with more than one counterpart, due to some *fk_by* buildings and larger complexes having multiple parts that all carry the name of the whole. However, this did not occur among the ten matched features.

5.2. Use cases

In the following sections, the usability of the transformed and linked data is demonstrated through example queries. Each query makes use of prefixes to denote the namespaces of the vocabularies used. They are shown in a combined list below. The viewer (Section 4.7) adds the whole list to the top of every query, even when only a subset is needed.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX bu-base: <http://inspire.ec.europa.eu/ont/bu-base#>
PREFIX bu-core2d: <http://inspire.ec.europa.eu/ont/bu-core2d#>
PREFIX scalevocab: <http://temp.gis.lu.se/ont/scalevocab#>
PREFIX geonames: <http://www.geonames.org/ontology#>
```

5.2.1. Supplementing a gazetteer entry with a detailed geometry

An entry in GeoNames is represented geometrically as WGS 84 coordinates encoded with the Basic Geo Vocabulary. This is a sufficient amount of information for many applications, e.g. when pointing to a feature's location or placing a label on a map. In other cases, a polygon geometry might be needed, e.g. for showing a building's footprint or making accurate geometric calculations. In this use case, a building feature with the name *Allhelgonakyrkan* (All Saints' Church) in GeoNames is looked up, but instead of visualizing it with its own associated geometry, a linked high-LoD polygon representation is requested as a substitution. In the following query, the desired GeoNames feature is stated by its associated name string ("Allhelgonakyrkan"). The feature (*?feature*) has been linked with an authoritative equivalent (*?building*) which has an unknown number of representations (*?geometry*). The representation with the highest possible LoD is of interest here, so the reference (default) geometry is requested for visualization (using INSPIRE terms). This yields one result, shown in Figure 16.

```
SELECT ?wkt WHERE {
    ?feature a geonames:Feature ;
        geonames:name "Allhelgonakyrkan" ;
        skos:closeMatch ?building .
    ?building a bu-core2d:Building ;
        bu-core2d:geometry2D ?geometry .
    ?geometry bu-base:BuildingGeometry2D.geometry ?geometryobj ;
        bu-base:BuildingGeometry2D.referenceGeometry "true"^^xsd:boolean .
    ?geometryobj geo:asWKT ?wkt .
}
```

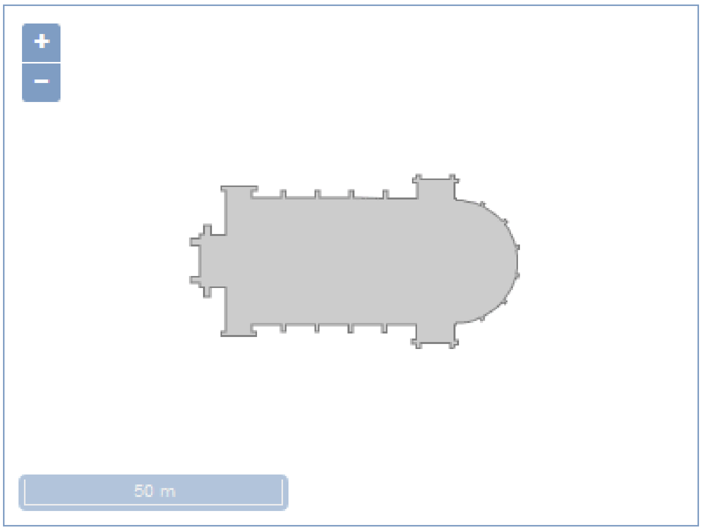


Figure 16. Screenshot of a building feature from GeoNames (Allhelgonakyrkan church in Lund) visualized as a high-LoD polygon.

5.2.2. Requesting suitable geometries for a given scale level

The most immediate application area for multiple representation on the Semantic Web is web mapping. With the authoritative representations annotated with scale information, the knowledge of which ones are suitable for display is contained within the data rather than in software code. In this use case, a suitable representation is retrieved and rendered based on a web map's zoom level. In the following query, each stored representation (*?geometry*) has an associated WKT literal (*?wkt*) and a scale (*?scale*). The viewer (Section 4.7) reads the map's zoom level and replaces any occurrence of "scaleFromZoom" in the query with an approximated scale value. Results are returned if this value falls within the scale's viewing bounds. The rendered geometries for three approximated scale levels are shown in Figure 17.

```
SELECT ?wkt WHERE {
    ?geometry a bu-base:BuildingGeometry2D ;
        bu-base:BuildingGeometry2D.geometry ?geometryobj ;
        scalevocab:hasScale ?scale .
    ?geometryobj geo:asWKT ?wkt .
    ?scale scalevocab:hasLowerBound ?lowerbound ;
        scalevocab:hasUpperBound ?upperbound .

    FILTER (?lowerbound <= scaleFromZoom && ?upperbound > scaleFromZoom) .
}
```

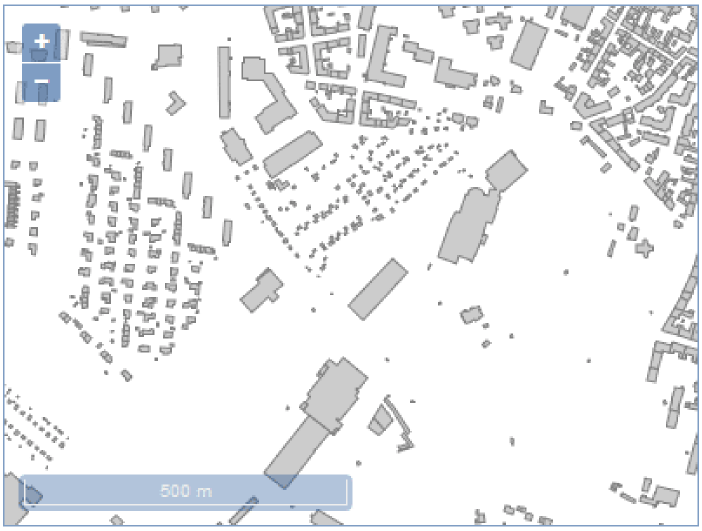
5.2.3. Viewing linked attribute information for a selected geometry

In the previous section, representation levels were retrieved and visualized in a web map, ending with a small-scale view. The geometries shown at this level are derived from the Road Map data (*vk_by*), which hold very few feature attributes. With multiple geometries linked to features, they share attributes that were previously isolated. In this use case, one attribute type, building use, is requested when a geometry is selected by the user. The viewer sends the following query to the triplestore (without re-rendering the map) after replacing any occurrence of "interiorPoint" with the point of the clicked location. If the point is spatially within a polygon (*?geometryobj*), the building use categories (*?resource*) of features (*?building*) represented by the polygon are returned as plain text (*?label*) and displayed in the browser. The generalized polygon selected in Figure 17c represents two large buildings and a few smaller ones. It therefore has several building uses, but each category is only returned once.

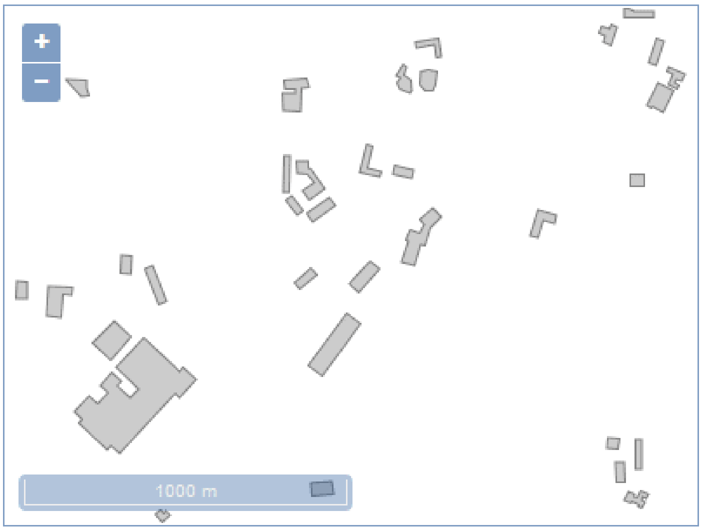
```
SELECT ?label WHERE {
    ?building a bu-core2d:Building ;
        bu-core2d:geometry2D ?geometry ;
        bu-base:AbstractBuilding.currentUse ?use .
    ?use bu-base:CurrentUse.currentUse ?resource .
    ?resource skos:prefLabel ?label .
    ?geometry a bu-base:BuildingGeometry2D ;
        bu-base:BuildingGeometry2D.geometry ?geometryobj ;
        scalevocab:hasScale ?scale .
    ?scale scalevocab:hasLowerBound ?lowerbound ;
        scalevocab:hasUpperBound ?upperbound .

    "interiorPoint"^^geo:wktLiteral geof:within (?geometryobj) .
    FILTER (?lowerbound <= scaleFromZoom && ?upperbound > scaleFromZoom) .
}
```

a)



b)



c)

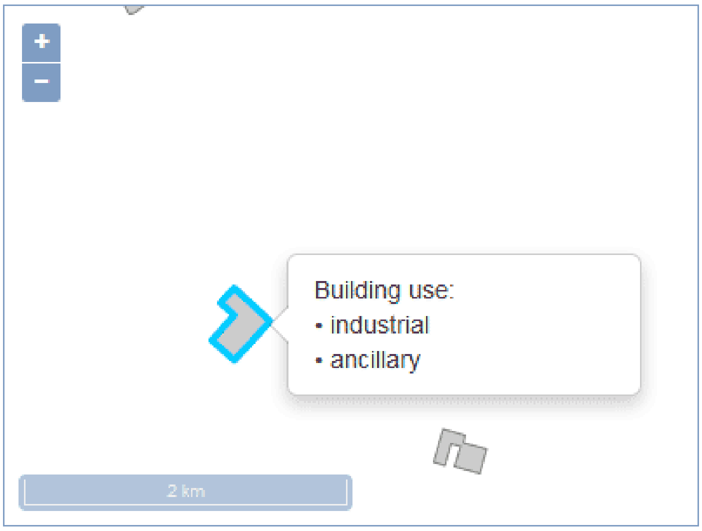


Figure 17. Screenshots of geometries retrieved based on zoom levels: a) 1:17 062, b) 1:34 124, and c) 1:68 247, together with linked attribute information.

6. Discussion

6.1. Authoritative multi-scale geodata

Regarding the first research question in Section 1.2, the case study shows that multi-scale national data sets can be transformed into RDF, but doing so requires careful planning and possibly a revisit of the data maintenance process. NMAs usually make updates within central databases that are used to compile products for distribution. So far, transformations into RDF have primarily been done using ad hoc vocabularies that simply mimic the original database schemas, but currently, there is a trend towards harmonization. The latter route is arguably more advantageous, but might need extensive work if the data model of the target vocabulary is radically different from that of the source data (such as the tabular data sets used in the case study). The object-oriented nature of GeoSPARQL and INSPIRE means that the data producer must first set up a system that incorporates their class structures so that the data can eventually be transformed. Most importantly, this involves the separation of features and geometries, as well as the creation of a unique ID for every single class instance rather than each feature record.

As for the enhancement of visualization, the use cases in Chapter 5 show that the linked multi-scale data can not only supplement related information on the Semantic Web (the GeoNames project in this case) with more detailed geometries than would otherwise be accessible, but also deliver appropriate geometric representations for display in web maps. Furthermore, interlinked geometries at multiple LoDs can point to associated attribute information of the features they represent. However, it should be emphasized that the use cases are very limited in scope; they do not cover all available feature- and geometry attributes, different ways of querying geometries, and more. The case study also shows that transforming and interlinking data sets at multiple LoDs requires some planning. Despite the possibility of matching geometries from separate data sets, it is preferable to include the linking process in the data production itself. The data producer must then decide how to organize the multiple representation environment before performing the transformation. Regardless of how the data are compiled, each stored representation should be provided with its own ID, as this allows it to be described with attributes such as LoD, accuracy, temporal version etc.

6.2. Geospatial vocabularies

Regarding the second research question in Section 1.2, it is evident that current geospatial vocabularies partly fulfil the needs of multi-scale national data sets. The data models behind those used in the case study are the result of extensive planning and consultation. They both make well-thought-out distinctions between related concepts and their relationships, which is ideal for transformation of data into RDF. The small yet encompassing GeoSPARQL vocabulary provides top-level classes and properties that can be extended with more detailed schemas (as has been done with the Simple Features vocabulary). The only problem encountered with its use is the mismatch between the standard and its official implementation regarding the predicate *geo:hasDefaultGeometry* (see Section 3.4.1). While some triplestores have built-in support based on the standard, others might simply have the implemented vocabulary imported, and as a result, users might handle the problem differently or not be aware of it. However, this is a minor oversight which will likely be corrected before the technology reaches wider adoption.

Although the INSPIRE vocabularies are work in progress, it is clear that the completed version will be an important tool for encoding any kind of geodata in RDF. The large number of themes, concepts, and attributes defined in the application schemas are a good fit for the task. For the few types of attributes taken from the authoritative building data in the case study, the mapped vocabularies worked very well. The separation of attributes relating to features on one hand and geometries on the other is also a useful quality that allows a single real-world object to have several geometric versions. However, there is no obvious way to allow several temporal versions of the geometries, something that would be of much value for future data publication.

Both vocabulary sets are designed to allow multiple representation of features, and each works well in isolation. However, the case study reveals two problems with how the current version of the INSPIRE vocabularies is integrated with GeoSPARQL in this regard. The first is that all properties linking features and their representations are defined as sub-properties of *geo:hasDefaultGeometry*, a relationship that is not found in the application schemas. When a feature has several geometries, a query written with GeoSPARQL terms will unexpectedly return all of them as default, even if the data producer has not marked them as such. The conceptual integration of the two vocabularies is thus broken. One solution would be to map all relevant properties to *geo:hasGeometry* and make use of *bu-base:BuildingGeometry2D .referenceGeometry* or its equivalent in other themes. The data producer can then decide whether to include an additional triple using *geo:hasDefaultGeometry* when applicable.

The other problem has to do with how geometry classes are mapped from the original schemas. In most cases, a feature class has a *GM_Object* as a property, which is realized in RDF as a feature having an *sf:Geometry* or equivalent. In some cases however, a special geometry class (a UML data type) is defined for holding various geometry attributes as well as the *GM_Object*. Currently, this is realized in RDF as a feature having a geometry and the geometry having an *sf:Geometry*. This presence of two classes for one concept means that the properties cannot inherit from GeoSPARQL without risking unexpected querying results. One solution would be to make *bu-base:BuildingGeometry2D* a subclass of *sf:Geometry* and *bu-core2d:geometry2D* a sub-property of *geo:hasGeometry* (Figure 18). This way, the subclass could hold relevant INSPIRE attributes and at the same time adhere to the GeoSPARQL model.

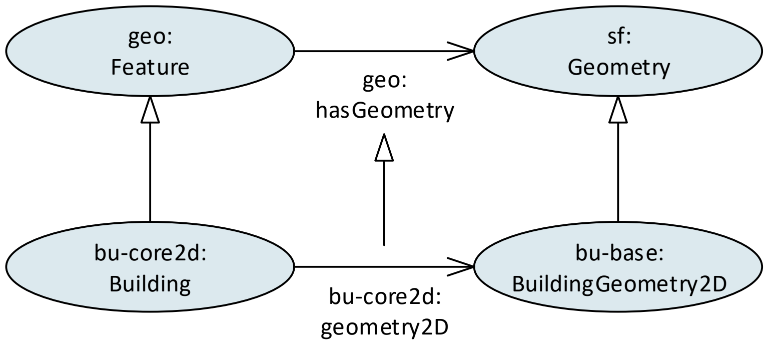


Figure 18. Possible improvements to the INSPIRE buildings vocabularies.

Despite both vocabulary sets allowing multiple geometries per feature, neither of them provides any standard way of telling them apart. The custom scale vocabulary created for the case study is not meant to be an ultimate solution, especially since the representative fraction is not well-suited for web applications. Nevertheless, having a special scale class to serve as an attribute of a geometry object proved to be very flexible and easy to work with, so this approach could well form the basis for a general LoD/resolution vocabulary in the future.

7. Conclusions

The objective of this thesis has been to assess the feasibility of bringing authoritative multi-scale geodata to the Semantic Web given the current technical situation, which mainly depends on whether Semantic Web technology is ready for multi-scale geodata and vice versa. To this end, Swedish national building data sets produced at three different scales were transformed into RDF and uploaded to a triplestore. They were encoded primarily using the GeoSPARQL vocabulary from the OGC and draft INSPIRE vocabularies that are currently being developed on behalf of the EC. Furthermore, the building data were linked to the GeoNames project to test integration of related information on the Semantic Web. In the absence of a shared vocabulary for expressing the scales of geometric representations, a custom prototype was implemented. Finally, the transformed, linked, and stored data were retrieved from the triplestore with the aid of a purpose-built viewer.

The presented case study consisted of initial inspection of the data and vocabularies used (Chapter 3), the process of transforming and linking the data (Chapter 4), and querying the output in three selected use cases (Chapter 5). It shows that authoritative multi-scale data sets can currently be brought to the Semantic Web, but moving forward with data publication in RDF using the chosen vocabularies should preferably be preceded by some groundwork. Among other things, this involves maintaining an object-oriented database according to the INSPIRE application schemas, creating a unique ID for every class instance, and deciding how the links between representation levels should be organized. The geospatial vocabularies used in the case study are, for the most part, suitable for the task at hand, but since they are yet to be completed and tested, it may be too early to proceed with extensive data publication at this time. In addition, the GI community will need to develop shared vocabularies for expressing LoD/resolution and other concepts that are not covered in those already under way.

References

Abele, A., McCrae, J.P., Buitelaar, P., Jentzsch, A., and Cyganiak, R., 2017. *Linking Open Data cloud diagram 2017*. [Online image] Available at <http://lod-cloud.net/> [Accessed 28 October 2017].

Atemezing, G.A., Abadie, N., Troncy, R., and Bucher, B., 2014. Publishing Reference Geodata on the Web: Opportunities and Challenges for IGN France. In: Kyzirakos, K., Gruetter, R., Kolas, D., Perry, M., Compton, M., Janowicz, K., and Taylor, K., eds. Proceedings of *6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web*, Riva del Garda, 20 October 2014. [Online] Available at <http://ceur-ws.org/Vol-1401/paper-01.pdf>.

Battle, R. and Kolas, D., 2012. Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web*, 3(4), pp.355–370.

Beek, W. and Folmer, E., 2017a. An Integrated Approach for Linked Data Browsing. In: Brovelli, M.A., Kotzinos, D., Paparoditis, N., and Raghavan, V., eds. Procedings (Academic track) of *FOSS4G-Europe 2017*, Marne La Vallée, 18–22 July 2017. ISPRS Archives XLII-4/W2. Göttingen: Copernicus. pp.35–38.

Beek, W., Folmer, E., Rietveld, L., and Walker, J., 2017b. GeoYASGUI: The GeoSPARQL Query Editor and Result Set Visualizer. In: Brovelli, M.A., Kotzinos, D., Paparoditis, N., and Raghavan, V., eds. Procedings (Academic track) of *FOSS4G-Europe 2017*, Marne La Vallée, 18–22 July 2017. ISPRS Archives XLII-4/W2. Göttingen: Copernicus. pp.39–42.

Berners-Lee, T., 1998. *Semantic Web Road map*. [Online] Available at <https://www.w3.org/DesignIssues/Semantic.html> [Accessed 27 March 2018].

Berners-Lee, T., 2006. *Linked Data*. [Online] Available at <https://www.w3.org/DesignIssues/LinkedData.html> [Accessed 27 March 2018].

Brickley, D., ed., 2003. *Basic Geo (WGS84 lat/long) Vocabulary*. W3C Semantic Web Interest Group. [Online] Available at <https://www.w3.org/2003/01/geo/> [Accessed 17 November 2017].

van den Brink, L., Janssen, P., Quak, W., and Stoter, J., 2014. Linking spatial data: semi-automated conversion of geo-information models and GML data to RDF. *International Journal of Spatial Data Infrastructures Research*, 9, pp.59–85.

Bulen, A., Carter, J.J., and Varanka, D.E., 2011. *A Program for the Conversion of The National Map Data from Proprietary Format to Resource Description Framework (RDF)*. Reston: USGS. [Online] Available at <https://pubs.usgs.gov/of/2011/1142/pdf/OFR2011-1142.pdf>.

Carral, D., Scheider, S., Janowicz, K., Vardeman, C., Krisnadhi, A.A., and Hitzler, P., 2013. An Ontology Design Pattern for Cartographic Map Scaling. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., and Rudolph, S., eds. Proceedings of *10th European Semantic Web Conference (ESWC 2013)*, Montpellier, 26–30 May 2013. Berlin/Heidelberg: Springer. pp.76–93.

Consoli, S., Gangemi, A., Nuzzolese, A.G., Peroni, S., Presutti, V., Reforgiato Recupero, D., and Spampinato, D., 2014. Geolinked Open Data for the Municipality of Catania. In: Akerkar, R., Bassiliades, N., Davies, J., and Ermolayev, V., eds. Proceedings of *4th International Conference on Web Intelligence, Mining and Semantics (WIMS'14)*, Thessaloniki, 2–4 June 2014. New York: ACM. No. 58.

Consoli, S., Presutti, V., Reforgiato Recupero, D., Nuzzolese, A.G., Peroni, S., Mongiovi, M., and Gangemi, A., 2017. Producing Linked Data for Smart Cities: The Case of Catania. *Big Data Research*, 7, pp.1–15.

Debruyne, C., Meehan, A., Clinton, É., McNerney, L., Nautiyal, A., Lavin, P., and O'Sullivan, D., 2017. Ireland's Authoritative Geospatial Linked Data. In: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., and Heflin, J., eds. Proceedings (part 2) of *16th International Semantic Web Conference (ISWC 2017)*, Vienna, 21–25 October 2017. Cham: Springer. pp.66–74.

Degbelo, A. and Kuhn, W., 2012. A Conceptual Analysis of Resolution. In: Namikawa, L.M. and Bogorny, V., eds. Proceedings of *XIII Brazilian Symposium on GeoInformatics (GEOINFO 2012)*, Campos do Jordão, 25–27 November 2012. pp.11–22.

Deniau, J., 2016. *Haversine*. Computer program. Version 0.4.5. [Online] Available at <https://pypi.python.org/pypi/haversine>.

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal of the European Union.

Egenhofer, M.J., 2002. Toward the Semantic Geospatial Web. In: Makki, K., Pissinou, N., Voisard, A., and Chen, S.C., eds. Proceedings of *10th ACM International Symposium on Advances in Geographic Information Systems (GIS '02)*, McLean, 8–9 November 2002. New York: ACM. pp.1–4.

European Commission, 2012. *Core Location Vocabulary*. Version 1.0. EC. [Online] Available at <https://joinup.ec.europa.eu/release/core-location-vocabulary/100>.

European Commission, 2013. *D2.8.III.2 INSPIRE Data Specification on Buildings – Technical Guidelines*. Version 3.0. EC. [Online] Available at <https://inspire.ec.europa.eu/sites/default/files/documents/data-specifications/inspire_dataspecification_bu_v3.0.pdf>.

European Commission, 2014. *D2.8.I.3 Data Specification on Geographical Names – Technical Guidelines*. Version 3.1. EC. [Online] Available at <https://inspire.ec.europa.eu/sites/default/files/documents/data-specifications/inspire_dataspecification_gn_v3.1.pdf>.

European Commission, 2016. *GeoDCAT-AP*. Version 1.0.1. EC. [Online] Available at <https://joinup.ec.europa.eu/release/geodcat-ap/v101>.

European Commission, n.d. a. *Agreed changes to the INSPIRE Technical Documentation for the "D2.8.I.2 INSPIRE data specification on Buildings – Technical Guidelines" version 3.0*. EC. [Online] Available at <https://inspire.ec.europa.eu/sites/default/files/corrigenda/corrigendum_on_buildings.pdf>.

European Commission, n.d. b. *INSPIRE code list register*. EC. [Online] Available at <http://inspire.ec.europa.eu/codelist>.

Flanagin, A.J. and Metzger, M.J., 2008. The credibility of volunteered geographic information. *GeoJournal*, 72(3–4), pp.137–148.

GeoNames, 2012. *GeoNames Ontology*. Version 3.1. [Online] Available at <http://www.geonames.org/ontology/documentation.html> [Accessed 8 March 2018].

GeoNames, 2017. *GeoNames Data*. [Online] Available at <http://www.geonames.org/export/> [Accessed 17 October 2017].

Goodchild, M.F. and Proctor, J., 1997. Scale in a Digital Geographic World. *Geographical & Environmental Modelling*, 1(1), pp.5–23.

Goodchild, M.F., 2007. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4), pp.211–221.

Goodwin, J., Dolbear, C., and Hart, G., 2008. Geographical Linked Data: The Administrative Geography of Great Britain on the Semantic Web. *Transactions in GIS*, 12(s1), pp.19–30.

Hahmann, S. and Burghardt, D., 2010. Linked Data - A Multiple Representation Database at Web Scale? In: Proceedings of *13th ICA Workshop on Generalisation and Multiple Representation*, Zürich, 12–13 September 2010. [Online] Available at <https://kartographie.geo.tu-dresden.de/downloads/ica-gen/workshop2010/genemr2010_submission_12.pdf>.

Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., and Thompson, H.S., 2010. When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., and Glimm, B., eds. Proceedings (part 1) of *9th International Semantic Web Conference (ISWC 2010)*, Shanghai, 7–11 November 2010. Berlin/Heidelberg: Springer. pp.305–320.

Hardy, P. and Field, K., 2012. Portrayal and Cartography. In: Kresse, W. and Danko, D.M., eds. *Springer Handbook of Geographic Information*. Berlin/Heidelberg: Springer. pp.323–358.

Hart, G. and Dolbear, C., 2013. *Linked Data: A Geographical Perspective*. Boca Raton: CRC Press.

Hietanen, E., Lehto, L., and Latvala, P., 2016. Providing Geographic Datasets as Linked Data in SDI. In: Halounova, L., Li, S., Šafář, V., Tomková, M., Rapant, P., Brázdil, K., Shi, W., Anton, F., Liu, Y., Stein, A., Cheng, T., Pettit, C., Li, Q., Sester, M., Mostafavi, M.A., Madden, M., Tong, X., Brovelli, M.A., Kang, H.K., Kawashima, H., and Coltekin, A., eds. Proceedings (Archives) of *XXIII ISPRS Congress, Commission II*, Prague, 12–19 July 2016. ISPRS Archives XLI-B2. Göttingen: Copernicus. pp.583–586.

Huang, W., Mansourian, A., and Harrie, L., 2016. On-demand mapping and integration of thematic data. In: Proceedings of *19th ICA Workshop on Generalisation and Multiple Representation*, Helsinki, 14 June 2016. [Online] Available at <https://kartographie.geo.tu-dresden.de/downloads/ica-gen/workshop2016/genemr2016_paper_02.pdf>.

Huang, W., Mansourian, A., Abdolmajidi, E., Xu, H., and Harrie, L., 2018. Synchronising geometric representations for map mashups using relative positioning and Linked Data. *International Journal of Geographical Information Science*, 32(6), pp.1117–1137.

Janowicz, K., Scheider, S., Pehle, T., and Hart, G., 2012. Geospatial semantics and linked spatiotemporal data – Past, present and future. *Semantic Web*, 3(4), pp.321–332.

Joint Research Centre, 2017a. *Guidelines for the RDF encoding of spatial data*. Draft. JRC. [Online] Available at <http://inspire-eu-rdf.github.io/inspire-rdf-guidelines/> [Accessed 19 December 2017].

Joint Research Centre, 2017b. *INSPIRE RDF Vocabularies*. Version 0.2. JRC. [Online] Available at <https://github.com/inspire-eu-rdf/inspire-rdf-vocabularies>.

Kadaster, 2016. *PDOK Data Platform – Datasets*. [Online] Available at <https://data.pdok.nl/datasets> [Accessed 19 February 2018].

Kilpeläinen, T., 2000. Maintenance of Multiple Representation Databases for Topographic Data. *The Cartographic Journal*, 37(2), pp.101–107.

Kolas, D., Hebeler, J., and Dean, M., 2005. Geospatial Semantic Web: Architecture of Ontologies. In: Rodríguez, M.A., Cruz, I., Egenhofer, M.J., and Levashkin, S., eds. Proceedings of *First International Conference on GeoSpatial Semantics (GeoS 2005)*, Mexico City, 29–30 November 2005. Berlin/Heidelberg: Springer. pp.183–194.

Kuhn, W., Kauppinen, T., and Janowicz, K., 2014. Linked Data - A Paradigm Shift for Geographic Information Science. In: Duckham, M., Pebesma, E., Stewart, K., and Frank, A.U., eds. Proceedings of *8th International Conference on Geographic Information Science (GIScience 2014)*, Vienna, 24–26 September 2014. Cham: Springer. pp.173–186.

Lantmäteriet, 2017a. *Lantmäteriets slutredovisning av uppdraget – Effekter och konsekvenser av öppna data*. Gävle: Lantmäteriet. [Online] Available at <https://www.lantmateriet.se/contentassets/870ebe20738f46d8a930f567a54f0030/109_2017_4287.pdf>.

Lantmäteriet, 2017b. *GSD-Fastighetskartan, vektor*. Digital data set.

Lantmäteriet, 2017c. *Produktbeskrivning: GSD-Fastighetskartan, vektor*. Version 7.3. Gävle: Lantmäteriet. [Online] Available at <https://www.lantmateriet.se/globalassets/kartor-och-geografisk-information/kartor/fastshmi.pdf>.

Lantmäteriet, 2017d. *GSD-Fastighetskartan, vektor*. Metadata. [Online] Available at <https://www.geodata.se/GeodataExplorer/GetMetaData?UUID=42f24def-0c5a-4ce4-a9db-b4a1e16f7fa6>.

Lantmäteriet, 2017e. *Handbok Byggnad*. Version 1.0. Gävle: Lantmäteriet. [Online] Available at <https://www.lantmateriet.se/globalassets/om-lantmateriet/var-samverkan-med-andra/kommunsamverkan/bal/handbocker/handbok_byggnad.pdf>.

Lantmäteriet, 2017f. *Frågor och svar - tjänstebaserad uppdatering*. [Online] Available at <http://www.lantmateriet.se/globalassets/om-lantmateriet/var-samverkan-med-andra/kommunsamverkan/bal/faq_fasad.pdf> [Accessed 28 December 2017].

Lantmäteriet, 2017g. *GSD-Terrängkartan, vektor*. Digital data set.

Lantmäteriet, 2017h. *Produktbeskrivning: GSD-Terrängkartan, vektor*. Version 5.1. Gävle: Lantmäteriet. [Online] Available at <https://www.lantmateriet.se/globalassets/kartor-och-geografisk-information/kartor/terrshmi.pdf>.

Lantmäteriet, 2017i. *GSD-Terrängkartan, vektor*. Metadata. [Online] Available at <https://www.geodata.se/GeodataExplorer/GetMetaData?UUID=a0e7d81e-9326-4233-8560-fbf6aa155938>.

Lantmäteriet, 2017j. *GSD-Vägkartan, vektor*. Digital data set.

Lantmäteriet, 2017k. *Produktbeskrivning: GSD-Vägkartan, vektor*. Version 4.1. Gävle: Lantmäteriet. [Online] Available at <https://www.lantmateriet.se/globalassets/kartor-och-geografisk-information/kartor/vagshmi.pdf>.

Lantmäteriet, 2017l. *GSD-Vägkartan, vektor*. Metadata. [Online] Available at <https://www.geodata.se/GeodataExplorer/GetMetaData?UUID=7f6776a0-471a-434e-aa71-fb5f6a79a01b>.

Lantmäteriet, 2018. *GSD-Sverigekartan 1:1 miljon, vektor*. Digital data set.

Lieberman, J., ed., 2006. *Geospatial Semantic Web Interoperability Experiment Report*. OGC. [Online] Available at <http://portal.myogc.org/files/?artifact_id=15198>.

Lieberman, J., Singh, R., and Goad, C., 2007. *W3C Geospatial Vocabulary*. W3C Geospatial Incubator Group. [Online] Available at <https://www.w3.org/2005/Incubator/geo/XGR-geo-20071023/> [Accessed 17 November 2017].

Mackaness, W.A. and Chaudhry, O., 2008. Generalization and Symbolization. In: Shekar, S. and Xiong, H., eds. *Encyclopedia of GIS*. New York: Springer. pp.330–339.

Montello, D.R., 2001. Scale in Geography. In: Smelser, N.J. and Baltes, P.B., eds. *International Encyclopedia of the Social & Behavioral Sciences*. Oxford: Pergamon Press. pp.13501–13504.

Open Geospatial Consortium, 2011. *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture*. Version 1.2.1. OGC. [Online] Available at <http://portal.opengeospatial.org/files/?artifact_id=25355>.

Open Geospatial Consortium, 2012a. *OGC GeoSPARQL - A Geographic Query Language for RDF Data*. Version 1.0. OGC. [Online] Available at <https://portal.opengeospatial.org/files/?artifact_id=47664>.

Open Geospatial Consortium, 2012b. *OGC® GeoSPARQL schema*. Version 1.0.1. OGC. [Online] Available at <http://schemas.opengis.net/geosparql/>.

Ordnance Survey, 2017. *Ordnance Survey Linked Data Platform*. [Online] Available at <http://data.ordnancesurvey.co.uk/> [Accessed 15 November 2017].

Östman, A. and Blomqvist, E., 2014. *Länkade Geodata - Omvärldsanalys*. s.l.: Projektet Länkade Geodata.

Patroumpas, K., Georgomanolis, N., Stratiotis, T., Alexakis, M., and Athanasiou, S., 2015. Exposing INSPIRE on the Semantic Web. *Web Semantics*, 35(p1), pp.53–62.

Ratte, J.B., 2016. *Pyjarowinkler*. Computer program. Version 1.8. [Online] Available at <https://pypi.python.org/pypi/pyjarowinkler>.

Salas, J.M., Harth, A., Norton, B., Vilches, L.M., de León, A., Goodwin, J., Stadler, C., Anand, S., and Harries, D., 2011. *NeoGeo Vocabulary: Defining a shared RDF representation for GeoData*. Draft. NeoGeo. [Online] Available at <http://geovocab.org/doc/neogeo.html> [Accessed 2 November 2017].

Sarjakoski, L.T., 2007. Conceptual Models of Generalisation and Multiple Representation. In:Mackaness, W.A., Ruas, A., and Sarjakoski, L.T., eds. *Generalisation of Geographic Information: Cartographic Modelling and Applications*. Kidlington: Elsevier. pp.11–35.

Schreiber, G., Raimond, Y., Manola, F., Miller, E., and McBride, B., eds., 2014. *RDF 1.1 Primer*. W3C Working Group Note. [Online] Available at <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>.

Shvaiko, P., Farazi, F., Maltese, V., Ivanyukovich, A., Rizzi, V., Ferrari, D., and Ucelli, G., 2012. Trentino Government Linked Open Geo-data: A Case Study. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., and Blomqvist, E., eds. Proceedings (part 2) of *11th International Semantic Web Conference (ISWC 2012)*, Boston, 11–15 November 2012. Berlin/Heidelberg: Springer. pp.196–211.

Stadler, C., Lehmann, J., Höffner, K., and Auer, S., 2012. LinkedGeoData: A core for a web of spatial open data. *Semantic Web*, 3(4), pp.333–354.

Tandy, J., van den Brink, L., and Barnaghi, P., eds., 2017. *Spatial Data on the Web Best Practices*. Working group note. W3C. [Online] Available at <https://www.w3.org/TR/2017/NOTE-sdw-bp-20170928/>.

Tirry, D., Vandenbroucke, D., Smith, R.S., and Lutz, M., 2015. *Guidelines on methodologies for the creation of RDF vocabularies representing the INSPIRE data models and the transformation of INSPIRE data into RDF*. JRC. [Online] Available at <https://joinup.ec.europa.eu/sites/default/files/document/2016-02/are3na-rdf_vocabulary_guidelines_final.pdf>.

Touya, G. and Brando-Escobar, C., 2013. Detecting Level-of-Detail Inconsistencies in Volunteered Geographic Information Data Sets. *Cartographica*, 48(2), pp.134–143.

Touya, G. and Reimer, A., 2015. Inferring the Scale of OpenStreetMap Features. In: Arsanjani, J.J., Zipf, A., Mooney, P., and Helbich, M., eds. *OpenStreetMap in GIScience: Experiences, Research, and Applications*. Cham: Springer. pp.81–99.

Tschirner, S., Scherp, A., and Staab, S., 2011. Semantic access to INSPIRE: How to publish and query advanced GML data. In: Grütter, R., Kolas, D., Koubarakis, M., and Pfoser, D., eds. Proceedings of *Terra Cognita Workshop on Foundations, Techologies and Applications of the Geospatial Web*, Bonn, 23 October 2011. [Online] Available at <http://ceur-ws.org/Vol-798/paper7.pdf>.

Usery, E.L. and Varanka, D.E., 2012. Design and development of linked data from The National Map. *Semantic Web*, 3(4), pp.371–384.

Vilches-Blázquez, L.M., Villazón-Terrazas, B., de León, A., Priyatna, F., and Corcho, O., 2010a. An Approach to Publish Spatial Data on the Web: The GeoLinked Data Case. In: Janowicz, K., Pehle, T., Hart, G., and Maué, P., eds. Proceedings of *Workshop on Linked Spatiotemporal Data 2010*, Zürich, 14 September 2010. [Online] Available at <http://ceur-ws.org/Vol-691/paper4.pdf>.

Vilches-Blázquez, L.M., Villazón-Terrazas, B., Corcho, O., and Gómez-Pérez, A., 2010b. GeoLinked Data: An application case. In: Proceedings of *I Jornadas Ibéricas de Infraestructuras de Datos Espaciales*, Lisbon, 27–29 October 2010. [Online] Available at <http://www.idee.es/resources/presentaciones/JIIDE10/ID473_GeoLinked_Data.pdf>.

Vilches-Blázquez, L.M., Villazón-Terrazas, B., Corcho, O., and Gómez-Pérez, A., 2014. Integrating geographical information in the Linked Digital Earth. *International Journal of Digital Earth*, 7(7), pp.554–575.

World Wide Web Consortium, 2004a. *OWL Web Ontology Language Reference*. W3C. [Online] Available at <https://www.w3.org/TR/2004/REC-owl-ref-20040210/>.

World Wide Web Consortium, 2004b. *XML Schema Part 2: Datatypes Second Edition*. W3C. [Online] Available at <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.

World Wide Web Consortium, 2009. *SKOS Simple Knowledge Organization System Reference*. W3C. [Online] Available at <https://www.w3.org/TR/2009/REC-skos-reference-2009 0818/>.

World Wide Web Consortium, 2013. *SPARQL 1.1 Query Language*. W3C. [Online] Available at <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/>.

World Wide Web Consortium, 2014a. *RDF 1.1 Concepts and Abstract Syntax*. W3C. [Online] Available at <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.

World Wide Web Consortium, 2014b. *RDF Schema 1.1*. W3C. [Online] Available at <https://www.w3.org/TR/2014/REC-rdf-schema-20140225/>.

World Wide Web Consortium, 2014c. *RDF 1.1 XML Syntax*. W3C. [Online] Available at <https://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/>.

World Wide Web Consortium, 2014d. *RDF 1.1 Turtle*. W3C. [Online] Available at <https://www.w3.org/TR/2014/REC-turtle-20140225/>.

Xu, H., 2017. *Development of a digitalization tool for linking thematic data to a background map*. Master's thesis. Lund University.

Zhang, X., Ai, T., Stoter, J., and Zhao, X., 2014. Data matching of building polygons at multiple map scales improved by contextual information and relaxation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92, pp.147–163.

Appendix A: Implementation of a scale vocabulary

The custom scale vocabulary designed for the study is shown below in Turtle format. See Section 4.2 for a description of its content. It may be used freely under CC0.

```
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
@prefix dc:         <http://purl.org/dc/elements/1.1/> .
@prefix skos:       <http://www.w3.org/2004/02/skos/core#> .
@prefix scalevocab: <http://temp.gis.lu.se/ont/scalevocab#> .

<http://temp.gis.lu.se/ont/scalevocab>
    a               owl:Ontology ;
    rdfs:label      "Scale vocabulary"@en ;
    dc:contributor  "E.K. Eiðsson", "W. Huang" ;
    dc:date         "2017-12-11"^^xsd:date ;
    owl:imports     <http://www.w3.org/2000/01/rdf-schema#> ,
    <http://www.w3.org/2002/07/owl#> , <http://www.w3.org/2001/XMLSchema#> ,
    <http://purl.org/dc/elements/1.1/> , <http://www.w3.org/2004/02/skos/core#> .

scalevocab:Scale
    a               owl:Class ;
    rdfs:label      "Cartographic scale"@en .

scalevocab:hasScale
    a               owl:ObjectProperty ;
    rdfs:label      "Scale of geometry object"@en ;
    rdfs:comment    "Domain is omitted to fit INSPIRE Bu vocabulary v0.2"@en ;
    rdfs:range      scalevocab:Scale .

scalevocab:hasReferenceLevel
    a               owl:DatatypeProperty ;
    rdfs:label      "Has reference level"@en ;
    rdfs:domain     scalevocab:Scale ;
    rdfs:range      xsd:float ;
    skos:definition "The denominator of the reference (default) level"@en .

scalevocab:hasUpperBound
    a               owl:DatatypeProperty ;
    rdfs:label      "Has upper bound"@en ;
    rdfs:domain     scalevocab:Scale ;
    rdfs:range      xsd:float ;
    skos:definition "The denominator of the least detailed level in range"@en .

scalevocab:hasLowerBound
    a               owl:DatatypeProperty ;
    rdfs:label      "Has lower bound"@en ;
    rdfs:domain     scalevocab:Scale ;
    rdfs:range      xsd:float ;
    skos:definition "The denominator of the most detailed level in range"@en .
```

Appendix B: Script for merging duplicate IDs

The following script was used to merge records sharing the same ID in the GSD-Property Map data set *BY*. See Section 4.3 (data preparation) for a description of its content.

```
# -*- coding: utf-8 -*-
""" Author: E.K. Eidsson; Version: 1.0; Date: 2018-01-12; Licence: BSD 2.0
    Merges records in Fastighetskartan BY that share the same OBJEKT_ID value. """

import arcpy
workspace = r"C:\budata"
arcpy.env.workspace = workspace

# Create a temporary geodatabase
arcpy.management.CreateFileGDB(workspace, "temp.gdb")

# Add a column for marking dissolved records
arcpy.management.AddField("fk_by.shp", "MARKER", "SHORT", )
arcpy.management.CalculateField("fk_by.shp", "MARKER", "1")

# Remove duplicate records
keepColumns = [["MARKER", "SUM"], ["OBJEKT_VER", "FIRST"], ["DETALJTYP", "FIRST"],
    ["ADAT", "FIRST"], ["INSAM_LAGE", "FIRST"], ["XYFEL", "FIRST"], ["NAMN1",
    "FIRST"], ["NAMN2", "FIRST"], ["NAMN3", "FIRST"], ["HUVUDBYGGN", "FIRST"],
    ["ANDAMAL_1", "FIRST"], ["ANDAMAL_1T", "FIRST"], ["ANDAMAL_2", "FIRST"],
    ["ANDAMAL_3", "FIRST"], ["ANDAMAL_4", "FIRST"], ["ANDAMAL_5", "FIRST"],
    ["ANDAMAL_6", "FIRST"], ["ANDAMAL_7", "FIRST"], ["ANDAMAL_8", "FIRST"],
    ["ANDAMAL_9", "FIRST"], ["ANDAMAL_10", "FIRST"]]
arcpy.management.Dissolve("fk_by.shp", "temp.gdb/fk_by", "OBJEKT_ID", keepColumns,
    "MULTI_PART")

# Restore column names
fields = arcpy.ListFields("temp.gdb/fk_by")
for field in fields:
    if field.name.startswith("FIRST_"):
        restoredName = field.name.replace("FIRST_", "")
        arcpy.management.AlterField("temp.gdb/fk_by", field.name, restoredName)

# Erase field values of merged records
unknownFields = ["SUM_MARKER", "OBJEKT_VER", "ADAT", "INSAM_LAGE", "XYFEL"]
table = arcpy.da.UpdateCursor("temp.gdb/fk_by", unknownFields)
for row in table:
    if row[0] > 1:
        row[1] = 0
        row[2] = "0"
        row[3] = "0"
        row[4] = 0
        table.updateRow(row)

# Replace input shapefile
arcpy.management.Delete("fk_by.shp")
arcpy.management.DeleteField("temp.gdb/fk_by", "SUM_MARKER")
arcpy.conversion.FeatureClassToShapefile("temp.gdb/fk_by", workspace)
```

Appendix C: Script for transforming buildings into RDF

The following script was used to transform authoritative building data from shapefiles into RDF form. See Section 4.4 (data transformation) for a description of its content.

```
# -*- coding: utf-8 -*-
""" Author: E.K. Eidsson; Version: 1.2; Date: 2018-02-27; Licence: BSD 2.0
    Reads shapefiles with building data at three levels of detail, transforms them
    into RDF, and writes linked features and geometries to a Turtle file. """

import arcpy
workspace = r"C:\budata"
arcpy.env.workspace = workspace

# Create a temporary geodatabase
if not arcpy.Exists("temp.gdb"):
    arcpy.management.CreateFileGDB(workspace, "temp.gdb")

# Transform shapefiles to WGS 84
wgs84 = arcpy.SpatialReference(4326)
arcpy.management.Project("fk_by.shp", "temp.gdb/fk_by84", wgs84,
    "SWEREF99_To_WGS_1984_1")
arcpy.management.Project("tk_by.shp", "temp.gdb/tk_by84", wgs84,
    "SWEREF99_To_WGS_1984_1")
arcpy.management.Project("vk_by.shp", "temp.gdb/vk_by84", wgs84,
    "SWEREF99_To_WGS_1984_1")

# Create an output file for triples
ttl = open("buildings_rdf.ttl", "w+")

# Add prefixes
ttl.write("@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .\n")
ttl.write("@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .\n")
ttl.write("@prefix geo: <http://www.opengis.net/ont/geosparql#> .\n")
ttl.write("@prefix sf: <http://www.opengis.net/ont/sf#> .\n")
ttl.write("@prefix basetypes: <http://inspire.ec.europa.eu/ont/base#> .\n")
ttl.write("@prefix gn: <http://inspire.ec.europa.eu/ont/gn#> .\n")
ttl.write("@prefix bu-base: <http://inspire.ec.europa.eu/ont/bu-base#> .\n")
ttl.write("@prefix bu-core2d: <http://inspire.ec.europa.eu/ont/bu-core2d#> .\n")
ttl.write("@prefix feature: <http://temp.gis.lu.se/bu/feature#> .\n")
ttl.write("@prefix fkgeom: <http://temp.gis.lu.se/bu/polygonGeom10k#> .\n")
ttl.write("@prefix tkgeom: <http://temp.gis.lu.se/bu/polygonGeom50k#> .\n")
ttl.write("@prefix vkgeom: <http://temp.gis.lu.se/bu/polygonGeom100k#> .\n")
ttl.write("@prefix fkgeomobj: <http://temp.gis.lu.se/bu/polygonGeomObj10k#> .\n")
ttl.write("@prefix tkgeomobj: <http://temp.gis.lu.se/bu/polygonGeomObj50k#> .\n")
ttl.write("@prefix vkgeomobj: <http://temp.gis.lu.se/bu/polygonGeomObj100k#> .\n")
ttl.write("@prefix scalevocab: <http://temp.gis.lu.se/ont/scalevocab#> .\n")
ttl.write("@prefix scale: <http://temp.gis.lu.se/shared/scale#> .\n")
ttl.write("@prefix name: <http://temp.gis.lu.se/bu/buildingName#> .\n")
ttl.write("@prefix spelling: <http://temp.gis.lu.se/bu/buildingNameSpelling#> .\n")
ttl.write("@prefix use: <http://temp.gis.lu.se/bu/buildingCurrentUse#> .\n\n")

```

```
# Create a fixed number of scale objects
ttl.write("scale:fk_10k a scalevocab:Scale .\n")
ttl.write('scale:fk_10k scalevocab:hasReferenceLevel "10000.0"^^xsd:float .\n')
ttl.write('scale:fk_10k scalevocab:hasUpperBound "20000.0"^^xsd:float .\n')
ttl.write('scale:fk_10k scalevocab:hasLowerBound "5000.0"^^xsd:float .\n')
ttl.write('scale:fk_10k rdfs:seeAlso <http://www.lantmateriet.se/sv/Kartor-och-
    geografisk-information/Kartor/Fastighetskartan/GSD-Fastighetskartan-vektor-/>
    .\n')
ttl.write("scale:tk_50k a scalevocab:Scale .\n")
ttl.write('scale:tk_50k scalevocab:hasReferenceLevel "50000.0"^^xsd:float .\n')
ttl.write('scale:tk_50k scalevocab:hasUpperBound "50000.0"^^xsd:float .\n')
ttl.write('scale:tk_50k scalevocab:hasLowerBound "25000.0"^^xsd:float .\n')
ttl.write("scale:tk_50k rdfs:seeAlso <http://www.lantmateriet.se/sv/Kartor-och-
    geografisk-information/Kartor/Terrangkartan/GSD-Terrangkartan-vektor/> .\n")
ttl.write("scale:vk_100k a scalevocab:Scale .\n")
ttl.write('scale:vk_100k scalevocab:hasReferenceLevel "100000.0"^^xsd:float .\n')
ttl.write('scale:vk_100k scalevocab:hasUpperBound "100000.0"^^xsd:float .\n')
ttl.write('scale:vk_100k scalevocab:hasLowerBound "50000.0"^^xsd:float .\n')
ttl.write("scale:vk_100k rdfs:seeAlso <http://www.lantmateriet.se/sv/Kartor-och-
    geografisk-information/Kartor/Vagkartan/GSD-Vagkartan-vektor/> .\n\n")

# Create unique IDs for tk and vk geometries
arcpy.management.AddField("temp.gdb/tk_by84", "GEOM_ID", "TEXT")
tk_by = arcpy.da.UpdateCursor("temp.gdb/tk_by84", ["GEOM_ID"])
counter = 1
for tkrow in tk_by:
    tkrow[0] = str(counter + 1000000)
    tk_by.updateRow(tkrow)
    counter += 1

arcpy.management.AddField("temp.gdb/vk_by84", "GEOM_ID", "TEXT")
vk_by = arcpy.da.UpdateCursor("temp.gdb/vk_by84", ["GEOM_ID"])
counter = 1
for vkrow in vk_by:
    vkrow[0] = str(counter + 1000000)
    vk_by.updateRow(vkrow)
    counter += 1

# Create geometry instances for tk and vk and assign scales
tk_by = arcpy.da.SearchCursor("temp.gdb/tk_by84", ["SHAPE@", "GEOM_ID"])
for tkrow in tk_by:
    tkGeomURI = "tkgeom:" + tkrow[1]
    ttl.write(tkGeomURI + " a bu-base:BuildingGeometry2D .\n")
    tkGeomObjURI = "tkgeomobj:" + tkrow[1]
    ttl.write(tkGeomObjURI + " a sf:Geometry .\n")
    ttl.write(tkGeomObjURI + ' geo:asWKT "' + tkrow[0].WKT + '"^^geo:wktLiteral
    .\n')
    ttl.write(tkGeomURI + " bu-base:BuildingGeometry2D.geometry " + tkGeomObjURI +
    " .\n")
    ttl.write(tkGeomURI + " scalevocab:hasScale scale:tk_50k .\n")
    ttl.write(tkGeomURI + ' bu-base:BuildingGeometry2D.referenceGeometry
    "false"^^xsd:boolean .\n\n')

```

```
vk_by = arcpy.da.SearchCursor("temp.gdb/vk_by84", ["SHAPE@", "GEOM_ID"])
for vkrow in vk_by:
    vkGeomURI = "vkgeom:" + vkrow[1]
    ttl.write(vkGeomURI + " a bu-base:BuildingGeometry2D .\n")
    vkGeomObjURI = "vkgeomobj:" + vkrow[1]
    ttl.write(vkGeomObjURI + " a sf:Geometry .\n")
    ttl.write(vkGeomObjURI + ' geo:asWKT "' + vkrow[0].WKT + '"^^geo:wktLiteral .\n')
    ttl.write(vkGeomURI + " bu-base:BuildingGeometry2D.geometry " + vkGeomObjURI + " .\n")
    ttl.write(vkGeomURI + " scalevocab:hasScale scale:vk_100k .\n")
    ttl.write(vkGeomURI + ' bu-base:BuildingGeometry2D.referenceGeometry "false"^^xsd:boolean .\n\n')

# Loop through records in fk and create features, geometries, and attributes
fk_by = arcpy.da.SearchCursor("temp.gdb/fk_by84", ["SHAPE@", "OBJEKT_ID", "OBJEKT_VER", "INSAM_LAGE", "XYFEL", "NAMN1", "NAMN2", "NAMN3", "ANDAMAL_1", "ANDAMAL_2", "ANDAMAL_3", "ANDAMAL_4", "ANDAMAL_5", "ANDAMAL_6", "ANDAMAL_7", "ANDAMAL_8", "ANDAMAL_9", "ANDAMAL_10"])
for row in fk_by:
    if row[3] != "4":
        # Add feature from fk to output file
        featureURI = "feature:" + row[1]
        ttl.write(featureURI + " a bu-core2d:Building .\n")

        # Add geometry from fk and assign scale
        fkGeomURI = "fkgeom:" + row[1] + "_" + str(row[2])
        ttl.write(fkGeomURI + " a bu-base:BuildingGeometry2D .\n")
        fkGeomObjURI = "fkgeomobj:" + row[1] + "_" + str(row[2])
        ttl.write(fkGeomObjURI + " a sf:Geometry .\n")
        ttl.write(fkGeomObjURI + ' geo:asWKT "' + row[0].WKT + '"^^geo:wktLiteral .\n')
        ttl.write(fkGeomURI + " bu-base:BuildingGeometry2D.geometry " + fkGeomObjURI + " .\n")
        ttl.write(fkGeomURI + " scalevocab:hasScale scale:fk_10k .\n")
        ttl.write(featureURI + " bu-core2d:geometry2D " + fkGeomURI + " .\n")
        ttl.write(featureURI + " geo:defaultGeometry " + fkGeomObjURI + " .\n")

        # Add attributes to geometry from fk
        ttl.write(fkGeomURI + ' bu-base:BuildingGeometry2D.referenceGeometry "true"^^xsd:boolean .\n')
        if row[3] == "3":
            ttl.write(fkGeomURI + " bu-base:horizontalGeometryReference <http://inspire.ec.europa.eu/codelist/HorizontalGeometryReferenceValue/" + "aboveGroundEnvelope> .\n")
        if row[4] != 0:
            ttl.write(fkGeomURI + ' bu-base:horizontalGeometryEstimatedAccuracy "' + str(row[4]) + '.0 mm"^^basetypes:Measure .\n')

        # Look for matching geometries from tk and vk
        tk_by = arcpy.da.SearchCursor("temp.gdb/tk_by84", ["SHAPE@", "GEOM_ID", "KKOD"])
        for tkrow in tk_by:
            if tkrow[2] == 690:
```

```
                if not tkrow[0].disjoint(row[0]):
                    if tkrow[0].intersect(row[0], 4).area >= (row[0].area / 2.0):
                        tkGeomURI = "tkgeom:" + tkrow[1]
                        tkGeomObjURI = "tkgeomobj:" + tkrow[1]
                        ttl.write(featureURI + " bu-core2d:geometry2D " + tkGeomURI + " .\n")
                        ttl.write(featureURI + " geo:hasGeometry " + tkGeomObjURI + " .\n")

    vk_by = arcpy.da.SearchCursor("temp.gdb/vk_by84", ["SHAPE@", "GEOM_ID", "KKOD"])
    for vkrow in vk_by:
        if vkrow[2] == 690:
            if not vkrow[0].disjoint(row[0]):
                if vkrow[0].intersect(row[0], 4).area >= (row[0].area / 2.0):
                    vkGeomURI = "vkgeom:" + vkrow[1]
                    vkGeomObjURI = "vkgeomobj:" + vkrow[1]
                    ttl.write(featureURI + " bu-core2d:geometry2D " + vkGeomURI + " .\n")
                    ttl.write(featureURI + " geo:hasGeometry " + vkGeomObjURI + " .\n")

    # Add feature attributes: building names from fk
    def addName(name, nameNo):
        if len(name) > 1:
            nameID = row[1] + nameNo
            ttl.write("name:" + nameID + " a gn:GeographicalName .\n")
            ttl.write(featureURI + " bu-base:AbstractConstruction.name name:" + nameID + " .\n")
            ttl.write("spelling:" + nameID + " a gn:SpellingOfName .\n")
            ttl.write("name:" + nameID + " gn:GeographicalName.spelling spelling:" + nameID + " .\n")
            ttl.write("spelling:" + nameID + ' gn:SpellingOfName.text "')
            ttl.write(name)
            ttl.write('"@sv .\n')

    addName(row[5].encode("utf-8"), "_1")
    addName(row[6].encode("utf-8"), "_2")
    addName(row[7].encode("utf-8"), "_3")

    # Add feature attributes: building uses from fk
    buildingUses = [row[8], row[9], row[10], row[11], row[12], row[13], row[14], row[15], row[16], row[17]]
    upperCategories = set([str(category)[0] for category in buildingUses])
    if "1" in upperCategories:
        ttl.write("use:" + row[1] + "_1 a bu-base:CurrentUse .\n")
        ttl.write(featureURI + " bu-base:AbstractBuilding.currentUse use:" + row[1] + "_1 .\n")
        ttl.write("use:" + row[1] + "_1 bu-base:CurrentUse.currentUse <http://inspire.ec.europa.eu/codelist/CurrentUseValue/residential> .\n")
    if "2" in upperCategories:
        ttl.write("use:" + row[1] + "_2 a bu-base:CurrentUse .\n")
        ttl.write(featureURI + " bu-base:AbstractBuilding.currentUse use:" + row[1] + "_2 .\n")
```

```
            ttl.write("use:" + row[1] + "_2 bu-base:CurrentUse.currentUse <http://inspire.ec.europa.eu/codelist/CurrentUseValue/industrial> .\n")
        if "3" in upperCategories or "4" in upperCategories:
            ttl.write("use:" + row[1] + "_3 a bu-base:CurrentUse .\n")
            ttl.write(featureURI + " bu-base:AbstractBuilding.currentUse use:" + row[1] + "_3 .\n")
            ttl.write("use:" + row[1] + "_3 bu-base:CurrentUse.currentUse <http://inspire.ec.europa.eu/codelist/CurrentUseValue/commerceAndServices> .\n")
        if "5" in upperCategories:
            ttl.write("use:" + row[1] + "_5 a bu-base:CurrentUse .\n")
            ttl.write(featureURI + " bu-base:AbstractBuilding.currentUse use:" + row[1] + "_5 .\n")
            ttl.write("use:" + row[1] + "_5 bu-base:CurrentUse.currentUse <http://inspire.ec.europa.eu/codelist/CurrentUseValue/agriculture> .\n")
        if "6" in upperCategories:
            ttl.write("use:" + row[1] + "_6 a bu-base:CurrentUse .\n")
            ttl.write(featureURI + " bu-base:AbstractBuilding.currentUse use:" + row[1] + "_6 .\n")
            ttl.write("use:" + row[1] + "_6 bu-base:CurrentUse.currentUse <http://inspire.ec.europa.eu/codelist/CurrentUseValue/ancillary> .\n")

        ttl.write("\n")
ttl.close()
```

Appendix D: Script for extracting GeoNames entries

The following script was used to extract GeoNames entries that fell within the study area. See Section 4.5 (data linking) for a description of its content.

```
# -*- coding: utf-8 -*-
""" Author: E.K. Eidsson; Version: 1.0; Date: 2018-02-03; Licence: BSD 2.0
    Reads a GeoNames text file and extracts records within a given bounding box. """

infile = open("SE.txt", 'r')
outfile = open("geonames_studyArea.txt", "w+")

# Extract records that fall within the study area (decimal degrees SNWE)
for line in infile:
    record = line.split("\t")
    if 55.679809 < float(record[4]) < 55.735098 and 13.144668 < float(record[5]) < 13.242590:
        outfile.write(line)

infile.close()
outfile.close()
```

Appendix E: Script for linking GeoNames with buildings

The following script was used to search for correspondence between GeoNames entries and building features, and then link them explicitly. See Section 4.5 (data linking) for a description of its content.

```
# -*- coding: utf-8 -*-
""" Author: E.K. Eidsson; Version: 1.0; Date: 2018-02-04; Licence: BSD 2.0
    Matches and links GeoNames records with building features.
    Requires <https://pypi.python.org/pypi/pyjarowinkler> and
    <https://pypi.python.org/pypi/haversine>. """

import arcpy
import math
from pyjarowinkler import distance
from haversine import haversine
workspace = r"C:\budata"
arcpy.env.workspace = workspace

# Create an output file for triples
ttl = open("geonames_rdf.ttl", "w+")

# Add prefixes
ttl.write("@prefix skos: <http://www.w3.org/2004/02/skos/core#> .\n")
ttl.write("@prefix feature: <http://temp.gis.lu.se/bu/feature#> .\n\n")

# Define functions
def s(a, b):
    jw0 = distance.get_jaro_distance(a[0].encode("utf-8"), b)
    jw1 = distance.get_jaro_distance(a[1].encode("utf-8"), b)
    jw2 = distance.get_jaro_distance(a[2].encode("utf-8"), b)
    return max([jw0, jw1, jw2])

def gc(a, b, c):
    d = haversine((a.Y, a.X), (float(b[0]), float(b[1])))
    if d > c:
        return 0.0
    else:
        return 1.0 / (1.0 + math.e**(-12.0 * (1.0 - d / c) + 6.0))

# Look for matches between fk and Geonames
fk_by = arcpy.da.SearchCursor("temp.gdb/fk_by84", ["SHAPE@", "OBJEKT_ID",
    "INSAM_LAGE", "NAMN1", "NAMN2", "NAMN3"])
for row in fk_by:
    if row[2] != "4" and len(row[3]) > 1:
        featureURI = "feature:" + row[1]

        geonamesData = open("geonames_studyArea.txt", "r")
        for line in geonamesData:
            record = line.split("\t")
            if record[6] == "S":
                s_ab = s([row[3], row[4], row[5]], record[1])
```

```
                gc_ab = gc(row[0].trueCentroid, [record[4], record[5]], 0.1)
                if (2.0/3.0) * s_ab + (1.0/3.0) * gc_ab > 0.95:
                    ttl.write(featureURI + " skos:closeMatch
    <http://sws.geonames.org/" + record[0] + "/> .\n")

        geonamesData.close()
ttl.close()
```

Series from Lund University

Department of Physical Geography and Ecosystem Science

Master's Thesis in Geographical Information Science

1. *Anthony Lawther:* The application of GIS-based binary logistic regression for slope failure susceptibility mapping in the Western Grampian Mountains, Scotland (2008).
2. *Rickard Hansen:* Daily mobility in Grenoble Metropolitan Region, France. Applied GIS methods in time geographical research (2008).
3. *Emil Bayramov:* Environmental monitoring of bio-restoration activities using GIS and Remote Sensing (2009).
4. *Rafael Villarreal Pacheco:* Applications of Geographic Information Systems as an analytical and visualization tool for mass real estate valuation: a case study of Fontibon District, Bogota, Columbia (2009).
5. *Siri Oestreich Waage:* a case study of route solving for oversized transport: The use of GIS functionalities in transport of transformers, as part of maintaining a reliable power infrastructure (2010).
6. *Edgar Pimiento:* Shallow landslide susceptibility – Modelling and validation (2010).
7. *Martina Schäfer:* Near real-time mapping of floodwater mosquito breeding sites using aerial photographs (2010).
8. *August Pieter van Waarden-Nagel:* Land use evaluation to assess the outcome of the programme of rehabilitation measures for the river Rhine in the Netherlands (2010).
9. *Samira Muhammad:* Development and implementation of air quality data mart for Ontario, Canada: A case study of air quality in Ontario using OLAP tool. (2010).
10. *Fredros Oketch Okumu*: Using remotely sensed data to explore spatial and temporal relationships between photosynthetic productivity of vegetation and malaria transmission intensities in selected parts of Africa (2011).
11. *Svajunas Plunge:* Advanced decision support methods for solving diffuse water pollution problems (2011).
12. *Jonathan Higgins:* Monitoring urban growth in greater Lagos: A case study using GIS to monitor the urban growth of Lagos 1990 - 2008 and produce future growth prospects for the city (2011).
13. *Mårten Karlberg:* Mobile Map Client API: Design and Implementation for Android (2011).
14. *Jeanette McBride:* Mapping Chicago area urban tree canopy using color infrared imagery (2011).
15. *Andrew Farina:* Exploring the relationship between land surface temperature and vegetation abundance for urban heat island mitigation in Seville, Spain (2011).
16. *David Kanyari*: Nairobi City Journey Planner: An online and a Mobile Application (2011).
17. *Laura V. Drews:* Multi-criteria GIS analysis for siting of small wind power plants - A case study from Berlin (2012).

18. *Qaisar Nadeem:* Best living neighborhood in the city - A GIS based multi criteria evaluation of ArRiyadh City (2012).
19. *Ahmed Mohamed El Saeid Mustafa:* Development of a photo voltaic building rooftop integration analysis tool for GIS for Dokki District, Cairo, Egypt (2012).
20. *Daniel Patrick Taylor*: Eastern Oyster Aquaculture: Estuarine Remediation via Site Suitability and Spatially Explicit Carrying Capacity Modeling in Virginia's Chesapeake Bay (2013).
21. *Angeleta Oveta Wilson:* A Participatory GIS approach to *unearthing* Manchester's Cultural Heritage '*gold mine'* (2013).
22. *Ola Svensson:* Visibility and Tholos Tombs in the Messenian Landscape: A Comparative Case Study of the Pylian Hinterlands and the Soulima Valley (2013).
23. *Monika Ogden:* Land use impact on water quality in two river systems in South Africa (2013).
24. *Stefan Rova:* A GIS based approach assessing phosphorus load impact on Lake Flaten in Salem, Sweden (2013).
25. *Yann Buhot:* Analysis of the history of landscape changes over a period of 200 years. How can we predict past landscape pattern scenario and the impact on habitat diversity? (2013).
26. *Christina Fotiou:* Evaluating habitat suitability and spectral heterogeneity models to predict weed species presence (2014).
27. *Inese Linuza:* Accuracy Assessment in Glacier Change Analysis (2014).
28. *Agnieszka Griffin:* Domestic energy consumption and social living standards: a GIS analysis within the Greater London Authority area (2014).
29. *Brynja Guðmundsdóttir:* Detection of potential arable land with remote sensing and GIS - A Case Study for Kjósarhreppur (2014).
30. *Oleksandr Nekrasov:* Processing of MODIS Vegetation Indices for analysis of agricultural droughts in the southern Ukraine between the years 2000-2012 (2014).
31. *Sarah Tressel:* Recommendations for a polar Earth science portal in the context of Arctic Spatial Data Infrastructure (2014).
32. *Caroline Gevaert:* Combining Hyperspectral UAV and Multispectral Formosat-2 Imagery for Precision Agriculture Applications (2014).
33. *Salem Jamal-Uddeen:* Using GeoTools to implement the multi-criteria evaluation analysis - weighted linear combination model (2014).
34. *Samanah Seyedi-Shandiz:* Schematic representation of geographical railway network at the Swedish Transport Administration (2014).
35. *Kazi Masel Ullah:* Urban Land-use planning using Geographical Information System and analytical hierarchy process: case study Dhaka City (2014).
36. *Alexia Chang-Wailing Spitteler:* Development of a web application based on MCDA and GIS for the decision support of river and floodplain rehabilitation projects (2014).
37. *Alessandro De Martino:* Geographic accessibility analysis and evaluation of potential changes to the public transportation system in the City of Milan (2014).
38. *Alireza Mollasalehi:* GIS Based Modelling for Fuel Reduction Using Controlled Burn in Australia. Case Study: Logan City, QLD (2015).

39. *Negin A. Sanati:* Chronic Kidney Disease Mortality in Costa Rica; Geographical Distribution, Spatial Analysis and Non-traditional Risk Factors (2015).
40. *Karen McIntyre:* Benthic mapping of the Bluefields Bay fish sanctuary, Jamaica (2015).
41. *Kees van Duijvendijk:* Feasibility of a low-cost weather sensor network for agricultural purposes: A preliminary assessment (2015).
42. *Sebastian Andersson Hylander:* Evaluation of cultural ecosystem services using GIS (2015).
43. *Deborah Bowyer:* Measuring Urban Growth, Urban Form and Accessibility as Indicators of Urban Sprawl in Hamilton, New Zealand (2015).
44. *Stefan Arvidsson:* Relationship between tree species composition and phenology extracted from satellite data in Swedish forests (2015).
45. *Damián Giménez Cruz*: GIS-based optimal localisation of beekeeping in rural Kenya (2016).
46. *Alejandra Narváez Vallejo:* Can the introduction of the topographic indices in LPJ-GUESS improve the spatial representation of environmental variables? (2016).
47. *Anna Lundgren:* Development of a method for mapping the highest coastline in Sweden using breaklines extracted from high resolution digital elevation models (2016).
48. *Oluwatomi Esther Adejoro:* Does location also matter? A spatial analysis of social achievements of young South Australians (2016).
49. *Hristo Dobrev Tomov:* Automated temporal NDVI analysis over the Middle East for the period 1982 - 2010 (2016).
50. *Vincent Muller:* Impact of Security Context on Mobile Clinic Activities: A GIS Multi Criteria Evaluation based on an MSF Humanitarian Mission in Cameroon (2016).
51. *Gezahagn Negash Seboka:* Spatial Assessment of NDVI as an Indicator of Desertification in Ethiopia using Remote Sensing and GIS (2016).
52. *Holly Buhler:* Evaluation of Interfacility Medical Transport Journey Times in Southeastern British Columbia. (2016).
53. *Lars Ole Grottenberg*: Assessing the ability to share spatial data between emergency management organisations in the High North (2016).
54. *Sean Grant:* The Right Tree in the Right Place: Using GIS to Maximize the Net Benefits from Urban Forests (2016).
55. *Irshad Jamal:* Multi-Criteria GIS Analysis for School Site Selection in Gorno-Badakhshan Autonomous Oblast, Tajikistan (2016).
56. *Fulgencio Sanmartín:* Wisdom-volkano: A novel tool based on open GIS and time-series visualization to analyse and share volcanic data (2016).
57. *Nezha Acil:* Remote sensing-based monitoring of snow cover dynamics and its influence on vegetation growth in the Middle Atlas Mountains (2016).
58. *Julia Hjalmarsson:* A Weighty Issue: Estimation of Fire Size with Geographically Weighted Logistic Regression (2016).
59. *Mathewos Tamiru Amato:* Using multi-criteria evaluation and GIS for chronic food and nutrition insecurity indicators analysis in Ethiopia (2016).
60. *Karim Alaa El Din Mohamed Soliman El Attar:* Bicycling Suitability in Downtown, Cairo, Egypt (2016).

61. *Gilbert Akol Echelai:* Asset Management: Integrating GIS as a Decision Support Tool in Meter Management in National Water and Sewerage Corporation (2016).
62. *Terje Slinning:* Analytic comparison of multibeam echo soundings (2016).
63. *Gréta Hlín Sveinsdóttir:* GIS-based MCDA for decision support: A framework for wind farm siting in Iceland (2017).
64. *Jonas Sjögren:* Consequences of a flood in Kristianstad, Sweden: A GIS-based analysis of impacts on important societal functions (2017).
65. *Nadine Raska:* 3D geologic subsurface modelling within the Mackenzie Plain, Northwest Territories, Canada (2017).
66. *Panagiotis Symeonidis*: Study of spatial and temporal variation of atmospheric optical parameters and their relation with PM 2.5 concentration over Europe using GIS technologies (2017).
67. *Michaela Bobeck:* A GIS-based Multi-Criteria Decision Analysis of Wind Farm Site Suitability in New South Wales, Australia, from a Sustainable Development Perspective (2017).
68. *Raghdaa Eissa*: Developing a GIS Model for the Assessment of Outdoor Recreational Facilities in New Cities Case Study: Tenth of Ramadan City, Egypt (2017).
69. *Zahra Khais Shahid*: Biofuel plantations and isoprene emissions in Svea and Götaland (2017).
70. *Mirza Amir Liaquat Baig*: Using geographical information systems in epidemiology: Mapping and analyzing occurrence of diarrhea in urban - residential area of Islamabad, Pakistan (2017).
71. *Joakim Jörwall*: Quantitative model of Present and Future well-being in the EU-28: A spatial Multi-Criteria Evaluation of socioeconomic and climatic comfort factors (2017).
72. *Elin Haettner*: Energy Poverty in the Dublin Region: Modelling Geographies of Risk (2017).
73. *Harry Eriksson*: Geochemistry of stream plants and its statistical relations to soil- and bedrock geology, slope directions and till geochemistry. A GIS-analysis of small catchments in northern Sweden (2017).
74. *Daniel Gardevärn:* PPGIS and Public meetings – An evaluation of public participation methods for urban planning (2017).
75. *Kim Friberg:* Sensitivity Analysis and Calibration of Multi Energy Balance Land Surface Model Parameters (2017).
76. *Viktor Svanerud:* Taking the bus to the park? A study of accessibility to green areas in Gothenburg through different modes of transport (2017).
77. *Lisa-Gaye Greene*: Deadly Designs: The Impact of Road Design on Road Crash Patterns along Jamaica's North Coast Highway (2017).
78. *Katarina Jemec Parker*: Spatial and temporal analysis of fecal indicator bacteria concentrations in beach water in San Diego, California (2017).
79. *Angela Kabiru*: An Exploratory Study of Middle Stone Age and Later Stone Age Site Locations in Kenya's Central Rift Valley Using Landscape Analysis: A GIS Approach (2017).
80. *Kristean Björkmann*: Subjective Well-Being and Environment: A GIS-Based Analysis (2018).

81. *Williams Erhunmonmen Ojo*: Measuring spatial accessibility to healthcare for people living with HIV-AIDS in southern Nigeria (2018).
82. *Daniel Assefa*: Developing Data Extraction and Dynamic Data Visualization (Styling) Modules for Web GIS Risk Assessment System (WGRAS) (2018).
83. *Adela Nistora*: Inundation scenarios in a changing climate: assessing potential impacts of sea-level rise on the coast of South-East England (2018).
84. *Marc Seliger*: Thirsty landscapes - Investigating growing irrigation water consumption and potential conservation measures within Utah's largest master-planned community: Daybreak (2018).
85. *Luka Jovičić*: Spatial Data Harmonisation in Regional Context in Accordance with INSPIRE Implementing Rules (2018).
86. *Christina Kourdounouli*: Analysis of Urban Ecosystem Condition Indicators for the Large Urban Zones and City Cores in EU (2018).
87. *Jeremy Azzopardi*: Effect of distance measures and feature representations on distance-based accessibility measures (2018).
88. *Patrick Kabatha*: An open source web GIS tool for analysis and visualization of elephant GPS telemetry data, alongside environmental and anthropogenic variables (2018).
89. *Richard Alphonce Giliba*: Effects of Climate Change on Potential Geographical Distribution of Prunus africana (African cherry) in the Eastern Arc Mountain Forests of Tanzania (2018).
90. *Eiður Kristinn Eiðsson*: Transformation and linking of authoritative multi-scale geodata for the Semantic Web: A case study of Swedish national building data sets (2018).