

Linking spatial data: semi-automated conversion of geo-information models and GML data to RDF*

Linda van den Brink¹, Paul Janssen¹, Wilko Quak^{1 2}, Jantien Stoter^{1 2 3}

¹ Geonovum; l.vandenbrink@geonovum.nl, p.janssen@geonovum.nl,
w.quak@geonovum.nl, j.stoter@geonovum.nl

² TU Delft; c.w.quak@tudelft.nl , j.e.stoter@tudelft.nl

³ Kadaster, Jantien.stoter@kadaster.nl

Abstract

Linked data provide an alternative route for the dissemination of spatial information compared to the traditional SOA-based SDI approach. The traditional approach has provided a wealth of standardized and structured location data based on Geography Markup Language (GML), while linked data provides an open mechanism for sharing and combining this data with anything, once the data is available as linked data. The first part of the paper focuses on deriving linked data from GML data. In the second part, we study how more meaningful data, expressed in Resource Description Framework (RDF) can be created from GML, given the underlying information model, by transforming it from Unified Modeling Language (UML) to Web Ontology Language (OWL).

Keywords: SDI, Linked Data, Semantic Web, GML, RDF

*This work is licensed under the Creative Commons Attribution-Non commercial Works 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

1. INTRODUCTION

Linked data provide an alternative route for the dissemination of spatial information compared to the traditional service-oriented architecture (SOA)-based Spatial Data Infrastructure (SDI) approach. Where the latter is built on predefined structuring of semantics within domains, linked data is open to linking information to any data over the Web. In this respect both are complementary: the traditional approach provides a mechanism for a basis of standardized and structured data within domains, while linked data provides an open mechanism for sharing and combining data. GML (Portele, 2012) as the ISO standard for exchange of service based spatial data and Resource Description Framework (RDF) (Klyne and Carroll, 2004) as the linked data format are therefore related. GML provides the format in which many spatial datasets are available and exchanged. This standardization process and effort has been realized on a large scale. The web of linked data could profit from this effort, as large amounts of standardized spatial information could be made available as linked data. This article will focus on the use of GML structured data as a source for deriving RDF structured data.

The first part of the paper focuses on deriving linked data from GML data. The first version of GML, v1.0 (Lake and Cuthbert, 2000), was based on RDF. From version 2.0 onwards GML was based on Extensible Markup Language (XML) and XML Schema, but the object-property structure was retained. We describe a transformation for translating any correctly structured GML to RDF automatically, using Extensible Stylesheet Language Transformations (XSLT) (Kay, 2007). Because GML's object-property structure translates very well to triples, the transformation is straightforward. Well-known GML content elements such as names and descriptions are mapped to their RDF equivalent. Geometries are transformed to Well Known Text (WKT), a compact format for expressing geometries described in Simple Features (ISO TC/211, 2004). However, any semantics specific to the input GML data (a.k.a. the application schema) are ignored in this translation.

In the second part, we study how more meaningful RDF can be created from GML, given the underlying information model, by transforming it from UML to Web Ontology Language (OWL) (W3C OWL working group, 2012). There exists a straightforward mapping to convert a UML model into an OWL vocabulary. However, the re-use of existing concepts in vocabularies takes a central role in OWL while in UML the use of vocabularies is not supported. We describe how annotating the UML model could improve this translation.

2. SPATIAL DATA AS REUSABLE WEB RESOURCES

Linked data is receiving increasing interest as a technology relevant for geographic information. It provides an alternative route for the dissemination of spatial

information compared to the already considered traditional SOA-based SDI approach. The key difference is that linked data is much more flexible and open. Where the SDI approach is built on predefined structuring of semantics within domains, linked data is open to linking information to any data over the Web. In this respect it much more appeals to the web 3.0 philosophy: unique information features that are available on the web and can be accessed and extended at any time, by anyone and for any purpose (CEN/TC287, 2012). In the Eye on Earth White Paper (Smits et al, 2011) the Linked Data approach and technologies are recommended for accessing resources in environmental information systems. However, the implication of this on what has been done and realized in the spatial information infrastructure until now is unclear. We argue that it is complementary. The traditional approach provides a mechanism for a basis of standardized and structured data within domains, while linked data provides an open mechanism for cross-domain sharing and combining. The traditional approach is characterized by a service based dissemination of GML structured data. In that approach data specifications provide clear definitions of semantics in predefined domains and use cases. These are implemented in XML schema, providing a well-defined and verifiable means of information exchange. The strong point of it is that the proper purpose of standardization and harmonization, interoperability, can be addressed through agreement and sharing of vocabulary. Once agreed the requirements and rules for communication are set and can be implemented in a verifiable way. The quality of implementation can be measured and therefore managed.

However, there is a downside: the semantics are defined within information domains. This results in predefined information silos, each related to different information domains. Within the context of a silo interoperability is assured by shared and foreseen concepts, but between silos little harmonization takes place, and for not yet foreseen concepts and relations the structure is too rigid. This is exactly the weak spot where linked data can be of help. It allows data to become part of a web of data where it is integrated with other data and data models can be interrelated and harmonized.

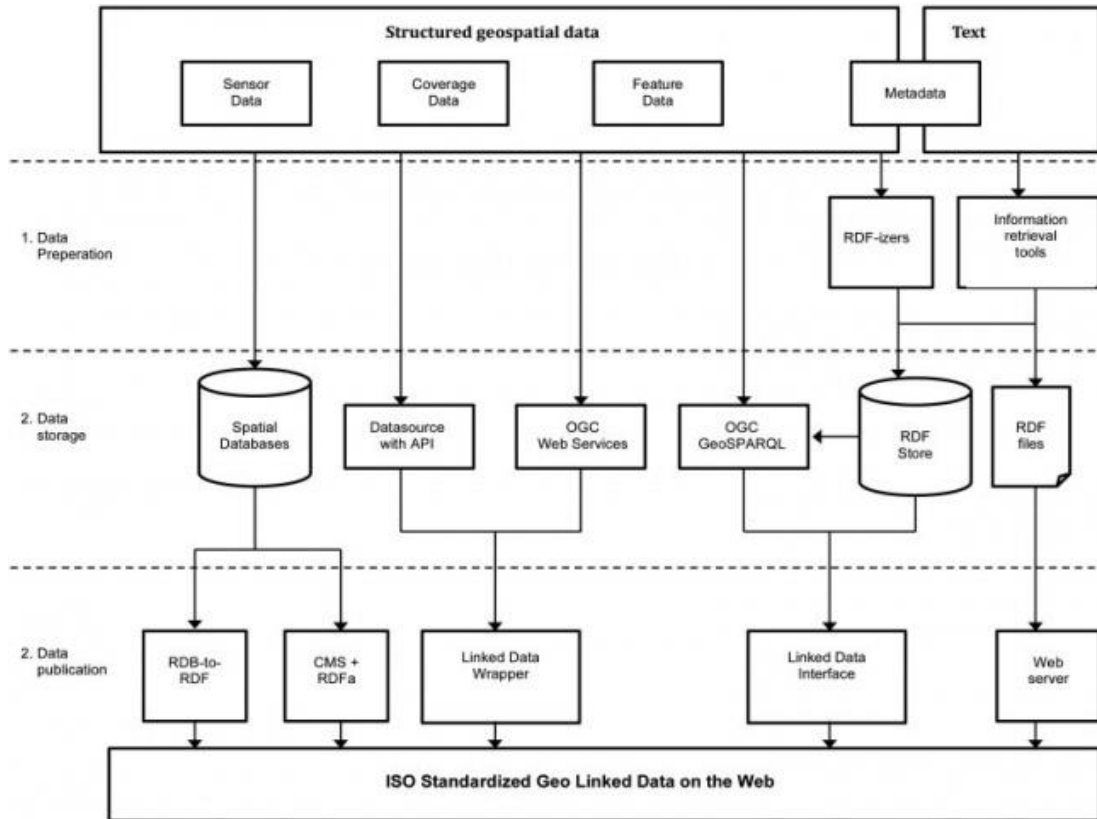
In a Spatial Data Infrastructure (SDI), GML data are generated and served from different feature based sources. Generally transformation services will do the transformation from these local sources to the GML structured data. In many SDI projects and programmes a lot of effort has been put into that activity. For linked data, including the related RDF format and the Geographic Query Language for RDF Data (GeoSPARQL, Perry and Herring, 2010), a similar approach can be followed: transformation services that act on the same local sources, generating linked data instead of GML, and making it available through GeoSPARQL endpoints. However, another option is to reuse the already existing GML sources. There are large amounts of existing feature data represented either in a GML file (or similar serialization) or in a data store supporting the general feature model (ISO/TC211, 2005). For example, the INSPIRE Directive of the European Union

(European Parliament, 2007) will lead to a lot of geospatial data becoming available in a standardized way. In addition, on the national level, e.g. in the Netherlands, more and more data is being published using Open Geospatial Consortium (OGC) standards. Since these data are already structured in a standardized and defined way, RDF transformation can be standardized as well. Linked data can therefore build on the structure already provided. One would expect a simple rule of benefit: profiting twice by reusing work that has been done once.

3. RELATED WORK

The problem of publishing spatial data as linked data is addressed in Schade and Cox (2010), who show that GML and RDF are isomorphic and suggest the addition of a façade on top of SDI web services to transform GML response data into RDF. Smits et al (2011) describe linked open data as a new paradigm for sharing resources and recommend RDF for making data available on the web. They suggest that linked data based approaches avoid the heavyweight platform, application, and interface issues inherent in many more traditional SDI approaches. López-Pellicer et al (2012) discuss principal ways of integrating linked data in a spatial data infrastructure. The following diagram from their publication depicts several ways of building linked data on top of an SDI:

Figure 1: RDF as Part of SDI



Source: López-Pellicer et al, 2012

As shown in this diagram, GML to RDF transformation can be positioned in the RDF-izer part. The Linked Data Wrapper is also of interest in this context. As mentioned, the idea was already described by Schade and Cox (2010). An implementation of such a wrapper, which creates linked data on top of OGC web services is described in G. Ciardi et al (2013). An ongoing effort to create spatial linked data sets is LinkedGeoData, an RDF representation of OpenStreetMap data, interlinked with other spatial data sets (Stadler et al, 2012).

A recent, thorough overview of ways to encode geometry in RDF can be found in Athanasiou et al (2013). This report was produced in the context of the GeoKnow project, a 3-year project funded by the European Commission. GeoKnow addresses the problem of linking geospatial data from heterogeneous information systems and uses the web as a platform to exploit this data (GeoKnow, 2013). Another relevant overview is given in the Core Vocabularies specification, which was developed as part of the Interoperability Solutions for European Public

Administration (ISA) programme of the European Union (ISA, 2012a). It has a section on location, the Location Core Vocabulary, which was recently installed in World Wide Web Consortium (W3C) space (Perego et al, 2013). The Location Core Vocabulary defines three classes, Location, Address, and Geometry, and a few properties for describing places in terms of their name, address or geometry.

An early effort to create and implement rules for transforming UML models into OWL ontologies was carried out in the context of an OGC Web Services Testbed (OWS-8) and is described in (Hobona and Brackin, 2011). Standardization efforts for such rules are currently underway within ISO/TC 211 project 19150-2 (ISO TC/211, 2013). This work will be published as an ISO Standard, and is currently in Draft International Standard (DIS) status. Experimentation with this mapping is being undertaken, for example by Cox (2013), who describes an explicit OWL representation of the ISO/OGC model for observations and measurements based on the ISO 19150-2 rules.

Tschirner et al (2011) achieved good results in transforming GML data models into OWL ontologies. We therefore used the experiences of these researchers and further develop the ideas and initial experiences into implementations to convert GML data to RDF instances accordingly. Also, we introduce more linked data geo vocabularies than GeoSPARQL to express geo data as linked data on the web and we provide, as main contribution, a method to link information models to existing vocabularies. New in our work is also the use of the above-mentioned standard ISO19150-2 that provides a set of rules to convert UML to OWL.

Another important topic in this context is how to create persistent, unique identifiers for data on the web. Developing a Uniform Resource Identifier (URI) strategy is one of the first steps needed to publish (geo) data as linked data. The W3C (and other experts groups) has led the development of guidelines and best practices on how to design "good" URIs in general and for the government domain in particular (W3C, 2012). In addition, several coordinating organizations issued guidelines on persistent URIs, such as the guidelines of ISA for European governments (ISA, 2012b), the guidelines for the UK public sector (Davidson, 2009) and the URI Design Principles of Linking Open Government Data (LOGD, 2013). Abbas and Ojo (2014) worked on the improvement of the implementation of such guidelines by consolidating existing URI design rules, distilling core URI design aspects or facets from these rules, and abstracting the rules into a set of consistent URI Design Patterns specifications. In our work we use the URI strategy designed by (Overbeek and van den Brink, 2013; Overbeek and Brentjens, 2013) that describe guidelines for formulating URIs for the Dutch public sector.

4. RESEARCH QUESTIONS AND METHOD

Geospatial data is already structured in a standardized and defined way, making it possible to standardize RDF transformation of this data as well. The GeoSPARQL standard specification recognizes this by mentioning the development of standard processes for converting (or virtually converting and exposing) these data to RDF as beneficial future work (Perry and Herring, 2010).

The challenge therefore is to investigate a way of generating linked data out of GML. Both GML and linked data have a level of data instances and a level of data model. In GML, data structures are defined in GML application schemata as extension of the abstract GML elements. Usually this data structure is expressed in a UML information model which defines the semantics and from which the GML application schema is automatically derived. In a linked data context, the data instances are expressed in RDF and the underlying model can be described as an ontology in OWL. Rules for transforming UML models into OWL ontologies are being developed within ISO/TC 211 project 19150-2 (ISO TC/211, 2013). There are several issues with the rules as described in this ISO standard, related to the closed world assumption of UML versus the open world assumption of linked data; the connection between concepts in a UML model and related concepts in vocabularies from the Semantic Web; and modelling conventions and restrictions in UML which are absent in OWL.

Several questions related to generating linked data out of GML are examined in this paper. First, is it possible to describe a generic transformation, without knowledge of the information model of the data, from GML to RDF? Second, how is geometry to be encoded in the resulting RDF? And third, how can GML application schemata, or the UML information models in which their structure and meaning are defined, be automatically transformed into an OWL representation, and integrated with other ontologies in the Semantic Web?

To address the first question, in section 4.1 we examine GML and to what extent it is similar to RDF. In section 4.2 we consider how geometry should be encoded in RDF by studying existing vocabularies and use cases. This gives us an overview of what options and criteria there are and allows us to select a method of encoding geometry. In section 4.3 we address the construction of URIs for our resources in RDF based on the Dutch URI strategy. The next step, which we describe in section 4.4, is to implement an experimental generic GML to RDF transformation in Extensible Stylesheet Language Transformations (XSLT) (Kay, 2007) and to test it with existing GML data. This allows us to determine if a generic GML to RDF transformation is possible and answers the first question. Note that the geometry2RDF tool (OEG, 2014) could have been an alternative to transform GML to RDF. However the source data of this tool is Oracle with geometry described in spatial columns and not in GML, as in our case.

Finally, in section 4.5 we describe our study of the third question by experimenting with an automated mapping from UML to OWL and with annotating the UML to better integrate the resulting ontology with existing ontologies or vocabularies on the web.

For the experiments, we use a sample GML file containing land use plans, conforming to the Dutch standard IMRO (Information Model Ruimtelijke Ordening - spatial planning), and the IMRO UML information model (Geonovum, 2012).

4.1. GML: A Triple Structure

GML (Geography Markup Language) is a standard for the storage and transport of geographic information. The first version of GML, v1.0 0, was published in May of the year 2000. Key concepts in the GML model of the world are the “feature”: an abstraction of a real world phenomenon; the “geographic feature”: a feature which is associated with a location relative to the Earth; and the “feature collection”, a collection of features which can itself be regarded as a feature and gives a digital representation of the real world. Features have properties; geographic features have properties whose value may be a geometry.

GML 1.0 used a geometry model called ‘Simple Features’, with definitions for point, line string, polygon, and some other basic geometric shapes. In addition it provided a coordinates element for encoding coordinates, and a Box element for defining extents. In its simplest form, GML contains no more semantics than this: geographic features with associated geometric shapes. The standard, however, includes an extension mechanism which makes it possible to define application-specific extensions with added semantics, for example distinct object classes for River and Road, each with their specific properties.

GML 1.0 described three encoding profiles for geographic features, two of which were based on XML, while the third was based on RDF and RDF Schema (RDFS). The model of GML was consistent with RDF. GML features have properties, which can have either literal values or have a geometry object as value (‘geometric value’). All features had an optional ID which could be used together with the GML document URI as a fragment identifier. In this manner, GML objects could be referenced as resources.

GML 1.0 example, in which 'yourhouse' and 'myhouse' have the same location:

```
<Building ID = 'yourhouse' .. >
  <location>
    <Point ID = '134'>
      <coordinates>
        2455.12, 3443.78
      </coordinates>
    </Point>
  </location>
</Building>
```

```
<Building ID = 'myhouse' .. >
  <location>
    <Point resource = '#134' />
  </location>
</Building>
```

From version 2.0 onwards GML was based on XML and XML Schema, and the RDF profile was no longer used. But an interesting fact is that the object-property structure, in which objects have properties and properties have either literal values or objects as values - basically a triple structure – always stayed, and is present in the current version, 3.3 0 (Portele, 2012).

4.2. Encoding Location in RDF

Resource Description Framework (RDF) (Klyne and Carroll, 2004) is a cornerstone standard of the Semantic Web. Where the World Wide Web as we know it is a web of interlinked documents, the Semantic Web is a web of interlinked data. RDF provides the foundation for publishing and linking data on the web.

RDF is slightly older than GML. The first version of the suite of standards documents related to RDF was published in February of 1999. A second version was published in 2004. RDF is to linked data what HyperText Markup Language (HTML) is to the World Wide Web: where HTML is a language for (re)presenting documents on the World Wide Web, RDF is a language for representing information about resources on the web. A resource can be a web document, about which metadata can be represented in RDF; the term 'resource' however can be interpreted more widely to mean anything that can be *identified* on the Web. This can be any (meta)data about anything, whether or not these things can be directly *retrieved* on the Web (Manola and Miller, 2004). In RDF, things are identified using Uniform Resource Identifiers (URIs) (Berners-Lee et al, 2005). Resources have properties and property values, property values can be URIs of other resources. RDF statements can be represented as a graph of nodes and arcs: the nodes represent the resources and values, the arcs represent the properties. The basic

resource - property - value structure of RDF can also be expressed as a triple consisting of subject, predicate, and object (always in that order). Each triple corresponds to a single arc plus the two nodes it connects in the graph.

As explained, RDF is a language for expressing any data, including geographic information, as resources. As a generic language, it has no specific features for encoding geometry, which is central to geographic information. However, several vocabularies and extensions have been proposed for this purpose. Athanasiou et al (2013) give an overview of currently available ways to encode a geometry in RDF and of RDF stores with geospatial support. The RDF stores are evaluated by measuring their performance with several geospatial queries against RDF triples.

When storing coordinates directly in RDF instead of referencing a spatial feature from another dataset, there are several options. These differ in what they offer, ranging from only lat/long point geometries, point line and surface geometries, topology, to the possibility to use any coordinate reference system. We give a short overview of the most well-known ones. A more extensive overview is given in Athanasiou et al (2013).

An early vocabulary for representing location data in RDF is W3C Basic Geo (Brickley, 2003). This vocabulary is explicitly created to be simple, providing just a few basic terms that can be used in RDF for describing lat(itude), long(itude) and other information about spatially-located things, using World Geodetic System 1984 (WGS84) as a reference datum. This RDF vocabulary was created with cross-domain data mixing in mind. By defining an encoding for point geometries in RDF, it makes it possible to “describe not only maps, but the entities that are positioned on the map.” (Brickley, 2003). Although a W3C activity, this vocabulary is not a W3C standard nor is it in the process of becoming one. As is evident from the name, the vocabulary is very basic and has only classes ‘geo:SpatialThing’ (similar to GML’s Feature) and ‘geo:Point’, and properties ‘geo:latitude’, ‘geo:location’, ‘geo:longitude’, and ‘geo:altitude’. It has no classes or properties for topology. Acceptance is high: it is used in both GeoNames and DBPedia, both in turn highly used data sets, and in web applications and services including Yahoo! Maps.

GeoRSS is an extension of Really Simple Syndication (RSS), a family of formats for news feeds on the web. GeoRSS adds the ability to encode geometries in RSS. It offers two methods for geometry encoding: Simple and GML. GeoRSS Simple is a very basic format with point, line, box and polygon properties, and allows only WGS84. GML is a GML application profile that supports a greater range of features and adds support for other coordinate reference systems. GeoRSS can be used in RDF to make simple geographical assertions about objects. However, this is only applicable for GeoRSS Simple, not for the GML variant. GeoRSS Simple is used

in e.g. DBPedia and implemented in e.g. OpenLayers, GeoServer, Drupal, and the Google Maps Application Programming Interface (API).

GeoSPARQL (Perry and Herring, 2010) is an OGC standard for representing and querying geospatial data on the Semantic Web. It defines a set of SPARQL Protocol and RDF Query Language (SPARQL) extension functions for spatial queries, a set of Rule Interchange Format (RIF) rules (Boley et al, 2013) for transforming simple topological relation tests into queries involving concrete geometries, and a core RDF/OWL vocabulary for geographic information. In this article the vocabulary is of interest, specifically where it addresses geometry. GeoSPARQL is based on accepted standards from the geospatial domain: the General Feature Model (ISO TC/211, 2005), Simple Features (ISO TC/211, 2004), Feature Geometry (ISO TC/211, 2003), and Structured Query Language Multimedia (SQL MM, ISO/IEC, 2011). It defines three classes, 'geosparql:SpatialObject', representing "anything that can have a spatial representation" [6, p. 6], and its subclass 'geosparql:Feature'; and 'geosparql:Geometry', representing the top-level geometry type. Also, several properties are defined for associating features with geometries and for recording metadata on geometries. In addition, it defines several sets of properties representing topological relations. The vocabulary allows two serializations for geometry, WKT and GML.

WKT is a text based format for encoding geometries, defined in the Simple Features specification. It is not only used in RDF, but is supported in several spatial databases and APIs. It is supported in several RDF semantic stores with geospatial capabilities, such as Virtuoso and uSeekM. The WKT option in GeoSPARQL allows only simple feature geometry types, but this is still a wide range of geometry types such as points, curves, surfaces and geometry collections. The GeoSPARQL vocabulary defines a property 'geosparql:asWKT' in which a geometry can be recorded as a text value. It is possible to use any coordinate reference system (CRS); a reference to the used CRS is recorded with the coordinates.

The GML option in GeoSPARQL allows all ISO 19107 spatial schema geometry types, which is a much wider range than the simple features allowed in WKT, including a lot of less commonly used types. The vocabulary defines a property 'geosparql:asGML' in which a geometry can be recorded as a GML literal, i.e. a geometry element from the GML schema can be embedded in the RDF. The 'geosparql:asGML' serialization is implemented in several tools (Athanasidou et al, 2013), and is offered as an option to record geometry in several vocabularies, such as the Location Core Vocabulary (part of the Core Vocabularies Specification, created as part of the European ISA programme). The Location Core Vocabulary (Perego et al, 2013) also allows WKT.

An alternative, more simple way of indicating a location in RDF is by referencing an already existing named place resource. DBPedia and GeoNames are existing RDF datasets with spatial features, i.e. besides place names and a lot of other information, a set of coordinates is also available for the objects in these datasets. The advantage of referring to these named place resources is that it makes clear that different resources which refer to, for example, <http://dbpedia.org/page/Utrecht>, are all referring to the same city. If these resources would not use a URI reference but a literal value "Utrecht" this could mean the province Utrecht, the city Utrecht (both places in the Netherlands), the South African town called Utrecht, or maybe something else entirely. In DBPedia and GeoNames (in the free service) however, spatial features are limited to the lat/long coordinates of a point, encoded using W3C Basic Geo. This means it is only possible to refer to a point on a map, which could be the location of a small object such as a traffic light, but also the centre point of a city or a country. These datasets do not contain, for example, the exact boundaries of a country, as this would be represented by a polygon feature instead of a single point.

Another ongoing effort to create spatial linked data sets is LinkedGeoData, an RDF representation of OpenStreetMap data (Stadler et al, 2012). This dataset contains nodes, ways and relations, representing all kinds of spatial features, such as roads or boundaries. The dataset is interlinked with DBpedia and GeoNames. Part of the effort is to establish a lightweight ontology based on OpenStreetMap, and an OWL vocabulary for the exchange and reuse of geographic data. LinkedGeoData uses points, lines and polygons to represent spatial features. For every way (such as a road), there exists a triple that contains the positions of all its nodes as a sequence. Each node is a point, represented as an OGC WKT literal. In addition the geometries of nodes are encoded as Basic Geo point geometries.

Since the IMRO test data uses the Dutch Rijksdriehoeksstelsel (RD) coordinate reference system and not WGS84, and uses both point and polygon geometry types, the best option is to use the GeoSPARQL vocabulary. For the experiments, WKT was selected because there was no need to go beyond simple feature geometry types and WKT is more compact than the GML serialization.

4.3. URI Strategy

An important question when creating RDF data is which criteria the URI identifiers which are formulated should follow and how to meet these criteria. In the Netherlands, we proposed a URI strategy describing guidelines for formulating URIs (Overbeek and van den Brink, 2013, Overbeek and Brentjens, 2013). The main criterion for linked data URIs is persistence: the URIs must still work even when things have changed, e.g. after the organization that minted them ceases to exist. For that reason a neutral domain name, with no organization name in it, is

preferred. Other criteria to take into account in a URI strategy are scalability, intelligibility, trust, machine-readability and human-readability.

The Dutch URI strategy describes the following URI pattern based on these criteria:

```
http://{domain}/{type}/{concept}/{reference}
```

```
{domain} = {internet domain}/{path}
```

This pattern is adopted from (ISA, 2012b). The `{domain}` serves two purposes. It is first of all an important instrument in obtaining unique identifications: two objects that are administered in two different databases can coincidentally be designated with the same identification. Should both objects be published as linked data, then two unique URIs will still be generated: both will start with a different internet domain name.

Secondly, a well-chosen domain will ensure recognisability and trust. Plots in the Land registry with a URI such as `http://data.brk.nl/perceel/010101` (BRK stands for the Dutch base registry for cadastral information) seem more reliable than `http://data.findithere.com/perceel/010101`, for example.

The 'persistence' criterion leads to two recommendations: the domain should be reserved exclusively for the publication of the register, not for other publications as well as this may result in a re-organisation of the publications now and then; and its name should not include the name of an organization as these tend to change over time.

The `{path}` can be used if various collections of objects exist within a register, in which double identifiers (IDs) may be present. The `{path}` can then be used to create extra namespaces. It is recommended to use this with restraint.

The `{type}` indicates which kind of URI is involved. This may be either 'id', 'doc', or 'def'. The first two are used to identify resources and information about these resources respectively, and the latter is used for ontology terms.

The `{concept}` part serves mainly to give the human reader an indication of the type of concept that is identified by the URI. In addition it offers a solution if there are objects within the registration that have no unique identifiers, but that are unique per type of object. One should also consider persistence when choosing the `{concept}`. If it is conceivable in a registration that object types (classes) can change names, while continuing to represent the same class, then it is not wise to include this component in the URI. A more abstract class should be used in such cases.

The `{reference}` is the identifying name or code of the individual object. This may be an identifying number, alphanumerical code, a word or name, etcetera.

In the case of IMRO, these recommendations on URI creation were applied in the following manner.

```
{domain} = http://data.ruimtelijkeplannen.nl
```

ruimtelijkeplannen.nl is the location on the web where Dutch spatial plans are publicly available. This is an appropriate, well-known and trustworthy domain name and not tied to a specific organization. Note that although we used this domain to assign URIs to our resources, no linked data is actually there in practice at this point in time. The optional `{path}` is not used.

```
{type} = 'id'
```

Only URIs of type 'id' are created. When information on the resource is supplied at the URI at a later stage, the type 'doc' will be used when supplying this information through content negotiation.

```
{concept} = 'ruimtelijkplan'
```

'ruimtelijkplan' (English: spatial plan) is a generic, conceptual name for the different types of plans that are described in the data. As the IMRO data model is subject to change every few years, and names of classes may change, we did not opt to use the class names of resources directly in the URI in the interest of persistence.

```
{reference} = [numeric id from the source data]
```

For the last part of the URI the numeric id of objects is taken from the source data. If a numeric id is not present, as is the case for nested, anonymous objects, an id is generated.

URIs for an IMRO vocabulary are also minted. These are created from the following pattern:

```
{domain} = http://data.ruimtelijkeplannen.nl
```

```
{type} = 'def'
```

This is followed by a hash '#' and the name of the vocabulary term. For the vocabulary, hash URIs are used to make the vocabulary recognisable as not part of the content; also the vocabulary is relatively small and can be requested as a whole in this way.

4.4. Experimental Transformation Implementation

Earlier in this article, we described GML in terms of its inherent triple structure. Because GML and RDF both have a triple structure, it is easy to define a transformation for translating any correctly structured (that is, conformant to the object-property triple structure) GML data to RDF automatically. As an experiment, we implemented such a transformation using XSLT 2.0. In this Generic-GML2RDF script, well-known GML content elements such as names and descriptions are mapped to their RDF equivalent. Objects, including nested features, data types and properties are recognized based on their place in the triple structure and are transformed accordingly.

The experimental implementation has 10 templates; counting whitespace and comments it has 98 lines. This shows the simplicity of the transformation. However, the stylesheet is presented here as a proof-of-concept and would need to be extended for general applicability. The stylesheet was tested on a sample GML file containing land use plans, conforming to the Dutch standard IMRO.

Figure 2: Sample XSLT Fragment

```

<xsl:template match="/">
  <rdf:RDF>
    <xsl:apply-templates select="//*[count(ancestor::* mod 2 = 0)
      [not(ancestor::schema-element(gml:_Geometry))]" />
  </rdf:RDF>
</xsl:template>

<xsl:template match="*[count(ancestor::* mod 2 = 0)
  [not(self::schema-element(gml:_Geometry))]">
  <rdf:Description rdf:about="{ $uridomain}/id/{local-name()}/
    {if (@gml:id) then @gml:id else generate-id(.)}"
    rdf:type="{ $uridomain}/def#{local-name()}">
    <xsl:apply-templates/>
  </rdf:Description>
</xsl:template>

<xsl:template match="gml:Point">
  <xsl:element name="sf:{local-name()}" rdf:about="{ $uridomain}/id/{local-name()}/
    {if (@gml:id) then @gml:id else generate-id(.)}">
    <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
      &lt;<xsl:value-of select="@srsName"/>>
      Point(<xsl:value-of select="gml:pos"/>)
    </geo:asWKT>
  </xsl:element>
</xsl:template>

<xsl:template match="gml:description">
  <rdfs:comment><xsl:value-of select="text()"/></rdfs:comment>
</xsl:template>

<xsl:template match="gml:name"><rdfs:label><xsl:value-of select="text()"/></rdfs:label>
</xsl:template>

```

The transformation starts at the top of the GML file and selects all features, even the ones that are nested as property value of another feature. The features can be recognized because they always have an even number of ancestors (levels in the XML hierarchy). The GML file starts with a feature (usually a feature collection), which has properties, which in turn have features as values. A simple filter can take advantage of this fact. Those elements that have an even number of ancestors (levels in the XML hierarchy) are transformed to `rdf:Description` elements. The `rdf:about` attribute is filled with `gml:id` if available; if not, an id is generated.

Well-known, standardized GML properties are transformed to an appropriate property. When possible a standard property from RDF or RDFS is used. For example, `gml:description` is transformed to `rdfs:comment`, `gml:name` to `rdfs:label`. Properties that are not known (i.e. not standard GML, but from the domain-specific IMRO extension) are not changed, but receive the same name in the RDF output.

In the IMRO data properties sometimes have nested content: they have an object as value, which is not referenced, but included in the hierarchical XML structure as a child element representing the object which in turn has child elements representing the properties. This construct receives special treatment in the stylesheet. The nested object is recognized by its even number of ancestors, and transformed to an `rdf:Description`. The property with nested content is transformed to a property that references the object that was nested, using an `rdf:resource` attribute containing the identifier of the feature prefixed with a hash '#'. Usually an identifier is not present in these cases, and one is generated automatically.

Properties that reference another feature in the GML data are transformed to an RDF ObjectProperty with an `rdf:resource` attribute containing the URI of the referenced feature.

The IMRO sample file contains simple Point and Surface geometries. These are transformed to a WKT serialization conform GeoSPARQL. The stylesheet currently only implements transformation of these two geometry types, but can easily be extended.

Figure 3: Sample IMRO RDF Fragment with WKT Geometry

```

<rdf:Description
  rdf:about="http://data.ruimtelijkeplannen.nl/id/Bouwaanduiding/NL.IMRO.0268.ID101733-00"
  rdf:type="http://data.ruimtelijkeplannen.nl/def#Bouwaanduiding">
  <imro:identificatie>NL.IMRO.0268.ID101733-00</imro:identificatie>
  <imro:typePlanobject>bouwaanduiding</imro:typePlanobject>
  <imro:plangebied rdf:resource="#NL.IMRO.0268.BP5000-VG01"/>
  <imro:naam>onderdoorgang</imro:naam>
  <imro:labelInfo rdf:resource="http://data.ruimtelijkeplannen.nl/id/Label/d34e17"/>
  <imro:geometrie rdf:resource="http://data.ruimtelijkeplannen.nl/id/Surface/d34e33"/>
</rdf:Description>
...
<sf:Surface rdf:about="http://data.ruimtelijkeplannen.nl/id/Surface/d34e33">
  <geo:asWKT rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">
    &lt;urn:ogc:def:crs:EPSG::28992&gt;
    Surface((187894.954 428671.17,187894.637 428671.314,187893.061 428672.029,
    187889.716 428673.547,187886.166 428665.573,187891.442
    428663.115,187894.954 428671.17))
  </geo:asWKT>
</sf:Surface>

```

The XSLT stylesheet described above transforms GML data to RDF in a generic way, based on GML's object-property structure. But it ignores any domain-specific semantics the GML data may have. The IMRO sample file has a lot of domain-specific semantics, defined in the IMRO GML application schema. As shown in Figure 3, `imro:Bouwaanduiding` (an element which gives rules about the external appearance of buildings in a planning area) is translated to an `rdf:Description` of `rdf:type http://data.ruimtelijkeplannen.nl/def#Bouwaanduiding`. All properties of `imro:Bouwaanduiding` are transformed to RDF properties of the same name (see Figure 3). These should all be defined in the IMRO ontology, which does not exist, at least not in RDF/OWL at this stage. Some of the properties could be mapped to Linked Data vocabularies. For example, it would be appropriate to translate `imro:naam` to `rdfs:label`, but this is not known within the transformation, as it is a generic tool and is not aware of the meaning of the IMRO vocabulary.

This aspect must be addressed, because usually GML is extended for a certain domain. It contains rich semantics, which would be lost in the translation to RDF. However, these semantics are of crucial importance in the context of the Semantic Web. Semantic extensions of GML are usually described in a standardized way in a well-documented UML model, from which the so-called GML application schema is automatically derived. For the Dutch IMRO standard such a UML model and application schema are available. Therefore not only the GML data, but also either the UML model or the GML application schema should be translated to Linked Data standards. In addition, domain-specific knowledge about the application schema could improve the mapping, taking into account established Linked Data languages and vocabularies like RDF and RDFS, as well as Dublin Core, Simple Knowledge Organization System (SKOS), or other vocabularies. In our experiments we looked

at this, and the next section describes some interesting aspects regarding the translation to RDF of specific semantics from an application-specific GML structure like IMRO.

4.5. Creating Meaningful RDF from Geo-Information Models

Meaning in the Semantic Web comes from vocabularies. However, the method described in the previous section does not provide or use a vocabulary. The mapping from GML to RDF would become more useful when it is accompanied by a OWL ontology for IMRO. Such an ontology can be automatically derived from the IMRO information model. The IMRO information model is available as UML diagram which, in current practice, is automatically converted into a GML application schema which supplies the rules for information exchange. This gives us two options to generate an OWL ontology: either from the UML model directly, or from the corresponding GML application schema. The first option has the advantage that UML is more mainstream information technology (IT) than GML application schemata and that a well-defined mapping from UML to OWL is defined by the OMG (Object Management Group, 2009). The second option (mapping from the GML application schema) has the advantage that it is spatially aware (since a GML application schema has well defined spatial semantics) which would result in a better mapping for spatial objects. A combination of both would be best; this can be achieved by defining specific mappings from UML for spatial modelling constructs. Currently these mappings are partially stable: for spatial datatypes a mapping is described in GeoSPARQL (Perry and Herring, 2010). How to map UML stereotypes used in spatial models (such as <<FeatureType>>) is still under development. Rules for transforming UML models into OWL ontologies, and rules for developing Application Schemas directly in OWL are currently being developed within ISO/TC 211 project 19150-2 (ISO TC/211, 2013). This work will be published as an ISO Standard, and is currently in Draft International Standard (DIS) status. Experimentation with this mapping is being undertaken, for example in Cox (2013).

4.5.1. UML to OWL mapping issues

There are several issues with the mapping as described in the ISO mapping that should be discussed.

The first issue, also described in Cox (2013), is related to the open-world assumption of OWL as opposed to the closed-world assumption of UML. ISO 19150-2 states that “OWL ontologies are complementary to UML static views and serve different purposes.” (p. vii) The OWL representation of UML models as described in ISO 19150-2 leans towards a more closed model: properties belong to one specific class, and any property that is not defined as belonging to a class, cannot be used with that class. Basically under the closed world assumption, anything that is not explicitly stated is not true within the closed system. By contrast, a popular saying to describe the open world assumption is that anyone can say

anything about anything. Properties are independent things that can be used to make a statement about a thing belonging to any class, and anyone can do this. Under an open world assumption, anything that is not explicitly stated is unknown.

Two options are available: either the OWL representation should as closely as possible reproduce the frame-based closed UML model from the standard, or it should embrace the open-world assumption of OWL. In the latter case the OWL model would be a representation of the underlying model, but without exactly reproducing the restrictions present in the UML model. This affects property scoping and object property restrictions. The latter choice requires more interpretation during conversion.

The second issue is that the mapping rules convert the UML diagram to an OWL ontology without linking the terms in this vocabulary to terms in existing OWL vocabularies. This may be acceptable in the closed world context of ISO standards, but in OWL interoperability is achieved via the re-use of vocabularies. Janowicz et al (2014) give vocabularies that reuse existing vocabularies 3 out of 5 stars in their proposed five star model for linked data vocabulary use (Janowicz et al, 2014).

The third issue is that some modelling conventions in UML (like the avoidance of multiple inheritance) can result in awkward UML constructions. When such a UML model gets mapped to OWL the awkwardness of the UML modelling is preserved, even in cases where these modelling conventions do not make any sense in the OWL domain and could be expressed in a better way in OWL, representing the real world more closely than is possible in UML. This, however, is probably not solvable in an automated way.

4.5.2. *Experimentation*

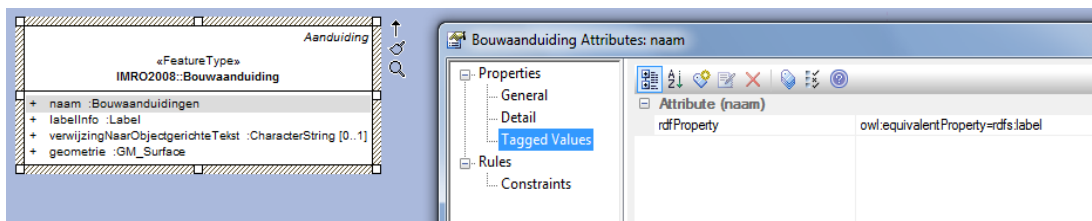
In this experiment we address the second issue, while also gaining more insight in the first and third issues. We propose to improve the mapping from UML to OWL by providing the missing information that is needed to create a better mapping. This is done by using tagged values in the UML model as mapping annotations which link terms from the IMRO model to terms from existing, well-known vocabularies. The transformation thus becomes a semi-automated process.

We experimented with these extended mapping rules by implementing them in an existing automated mapping tool: ShapeChange. ShapeChange implements an experimental, earlier set of UML to OWL mapping rules, described in (Hobona and Brackin, 2011). Feature types are mapped to owl:Class, their properties to owl:DatatypeProperty (for properties with literal values) or owl:ObjectProperty (for relations). Property names in the ontology are composed of the name of the UML class they belong to and their own name. Furthermore, rdfs:domain and rdfs:range are used to associate properties to their class. Codelists and enumerations are mapped to SKOS representations of the codelist.

In the first step of our experimentation we left the original mapping to OWL intact. The automated UML to OWL transformation was successfully applied to the IMRO model resulting in a IMRO OWL vocabulary. By slightly adapting the Generic-GML2RDF script it is possible to generate IMRO RDF from the GML data that refers to the IMRO vocabulary.

However, the IMRO vocabulary which was automatically generated by ShapeChange in this first step, is not harmonized with existing RDF vocabularies. For example, it would be meaningful to map `imro:naam` (name) to `rdfs:label`. However, the knowledge that `imro:naam` in fact has the same semantics as `rdfs:label` is not available in the UML model and cannot be automatically mapped. In order to improve the UML model for better mapping to RDF we extend the UML model by annotating the UML attributes that have a matching, standardized meaning in some well-known vocabulary in RDF, with a link to their RDF counterpart. Classes can be annotated in the same way. The annotation is recorded in UML via a tagged value. For example, the `imro:naam` attribute in the UML model is given the following annotation: `'owl:equivalentProperty=rdfs:label'`. The enrichment of the UML model with the mapping of IMRO classes and properties to existing OWL classes and properties with `equivalentProperty` and `equivalentClass` statements solves the second issue of the provided ISO mapping.

Figure 4: IMRO Bouwaanduiding with Annotation in UML



We extended ShapeChange with a few lines of code to transform this annotation to a triple stating that `imro:naam` is an equivalent property to `rdfs:label`.

Figure 5: IMRO Bouwaanduiding Vocabulary Entry as Generated by Shapechange

```
<DatatypeProperty xmlns="http://www.w3.org/2002/07/owl#" rdf:about="http://www.geonovum.nl/imro2008#Bouwaanduiding.naam">
  <owl:equivalentProperty rdf:resource="rdfs:label"/>
  <domain xmlns="http://www.w3.org/2000/01/rdf-schema#" rdf:resource="http://www.geonovum.nl/imro2008#Bouwaanduiding"/>
  <range xmlns="http://www.w3.org/2000/01/rdf-schema#" rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
  <definition xmlns="http://www.w3.org/2004/02/skos/core#" rdf:resource="http://www.geonovum.nl/imro2008#Bouwaanduidingen"/>
</DatatypeProperty>
```

This is a very simple example, but the implication is that a national standard such as IMRO, which in the Dutch context has very well-defined and agreed upon semantics, can now be mapped to international standards and thus be understood in an international context. If, for example, the INSPIRE data specifications were

derived as OWL ontology from their UML source, national standards could easily be related to INSPIRE, thus becoming part of a big, international web of meaning.

4.5.3. *Analysis of Results*

The resulting IMRO OWL model is improved with links to terms from existing, well-known vocabularies. In this respect, the experiment is a success. It requires manual annotation of the UML model, based on which the UML to OWL transformation can be automated. Thus, the experiment provides a semi-automated method for UML to OWL transformation. However, the other issues with UML to OWL mapping may prevent the semi-automatic creation of a useful OWL model from UML.

ShapeChange produces a closed-world oriented representation of the UML in OWL: by this we mean it reproduces as closely as possible the frame-based, closed UML model. We prefer the open-world oriented representation described by Cox (Cox, 2013), but nevertheless used ShapeChange in this experiment because it provides an automated approach which we could extend to support mapping annotations in the UML model. To be able to create an open-world oriented ontology as representation of a UML model, an open-world mapping should be defined. We suggest that ShapeChange can be modified to leave out the `rdfs:domain` and `rdfs:range` statements; these are logical axioms that 'close' the knowledge model. In addition, to make properties independent of classes, the class name should not be part of the property name (contrary to the rule in ISO 19150-2 par. 6.2.6).

In IMRO, properties that have the same name but belong to different UML classes can be treated as the same, as in each case the semantics of the property are the same. This was ascertained on a case-by-case basis. With an adapted mapping rule, which does not add domain and range to properties and does not use the class name in the property name, multiple properties with the same name would be generated in the UML to OWL transformation. The duplicates must either be removed by hand or with some automated method.

In other models, properties with the same name that belong to different UML classes may have class-dependent semantics, in which case they cannot be considered the same. For these cases, the proper mapping to a class-specific property in OWL, i.e. with the domain specified and a class-specific property name, can be configured in a UML annotation, which indicates that the property is class-specific.

Finally, whether a closed-world or open-world mapping is most applicable for a specific UML model can be decided when performing the mapping. The choice of open-world or closed-world mapping could be recorded on the package level or the class level by, for example, adding a tagged value `'rdfMappingRule= open'`.

It is unlikely that the third issue, UML anomalies getting mapped to RDF instead of being fixed, can be fixed by extending the UML model. IMRO is not a complex model and UML anomalies were not encountered during the experiment. To examine this in other cases, the generated OWL model would have to be analysed, providing conclusions and recommendations for each case.

An example of a UML anomaly is the avoidance of multiple inheritance. In UML models multiple inheritance is allowed, but often viewed as unwanted and therefore avoided. A result of this is often that classes have a number of properties that are exactly the same (often hard-copied from one class to the next). When this UML model gets mapped to OWL the awkwardness of the UML modelling is preserved (i.e. duplicate properties are created while there could have been just one). The adapted mapping rule, which maps properties with the same name in UML to just one property in OWL, would solve this.

Another example are the UML modelling constructs aggregation and composition (Atkinson and Kiko, 2008), for modelling mereological relationships. A similar construct is not available in OWL. In addition, a construction like the UML association class is not directly available in OWL. Atkinson and Kiko (2008) suggest that a common solution for representing n-ary and association class relations in OWL is reification, i.e., the creation of an individual, which stands for an instance of the relation and relates the things that are involved in that instance of the relation. Aggregation, composition, and association classes were not encountered in the IMRO model.

More experimentation with automatically creating an open world oriented OWL model from UML, as well as with more complex UML models containing modelling artefacts such as composition and association classes, is needed. Our experiment results in a working semi-automated method for mapping terms from UML models to existing OWL vocabularies, but does not address the other issues with UML to OWL transformation. This should be addressed in future work.

1.1. Source code availability

The IMRO2012 UML model, sample GML file containing IMRO spatial plans, result XML/RDF, genericGML2RDF XSLT, and adapted ShapeChange version and configuration file are downloadable from: <http://www.pilod.nl/wiki/BrinkEtAl-GML2RDF-files>.

The downloadable archive contains the source UML and GML files used in this experiment, the generic-GML2RDF XSLT stylesheet, and the adapted ShapeChange source and configuration file.

5. CONCLUSIONS AND FUTURE WORK

Linked data is complementary to GML published data. GML is advantageous for a controlled publication of data according to predefined semantic data specifications, while linked data facilitates an open publication environment in which additional information and data from other sources can easily be added. In this paper we demonstrated that it is possible to transform any GML to RDF using generic transformation rules, and without knowing the data model. OWL ontologies can be automatically derived from the original data specifications in UML. Once the data specification is available as an OWL ontology, it can be augmented with semantic relations to classes and properties from other ontologies, thus contributing to cross domain semantic harmonization. We introduced and implemented a method to annotate the data model in UML for this purpose. Deriving an OWL ontology from a UML model thus becomes a semi-automated process: manual annotation of the UML model with semantic annotations, followed by automatic transformation to OWL.

The implication of these results is that existing geo data can now be published on a large scale as semantically rich linked data. Linked data publication can be integrated in a SOA/GML based architecture, thus taking advantage of a large amount of geospatial data already being made available in national geoportals or, for example, in the context of INSPIRE. The IMRO information model and other Dutch information models could be derived as OWL ontologies, enriched with semantic relations to other ontologies from the Semantic Web, using the annotation method we described. It would also be beneficial to annotate the INSPIRE UML data models with mappings to existing, relevant ontologies and then to derive an INSPIRE ontology as well. The ontologies can then be interlinked and better harmonized, and INSPIRE data can be made available as linked data. In the Netherlands, the next step is to extend our existing SDI with the option to create linked data. The web of linked data can then take advantage of the existing wealth of standardized, open geospatial data.

Future work includes the implementation of ISO 19150-2 in an automated mapping tool such as ShapeChange and the addition of an option for creating open world-oriented ontologies. The problem of modelling conventions and restrictions in UML which lead to awkward modelling constructs, and could be better modelled differently in OWL, must still be addressed. Whether this is possible in an automated way as well remains an open question and seems, in some cases, unlikely.

REFERENCES

Abbas, S. and A. Ojo (2014), "Applying Design Patterns in URI Strategies -- Naming in Linked Geospatial Data Infrastructure," *Proceedings of the 47th*

Hawaii International Conference on System Sciences (HICSS), 6-9 Jan. 2014, Waikoloa, HI, pp. 2094-2103.

- Athanasiou, S., Bezati, L., Giannopoulos, G., Patoumpas, K. and Skoutas, D. (2013) *GeoKnow - Making the Web an Exploratory for Geospatial Knowledge. Market and Research Overview*. at http://svn.aksw.org/projects/GeoKnow/Public/D2.1.1_Market_and_Research_Overview.pdf [accessed 11 February 2014]
- Atkinson, C. and K. Kiko, (2008) A Detailed Comparison of UML and OWL. Reihe Informatik TR-2008-004
- Berners-Lee, T., Fielding, R. and L. Masinter (2005) Uniform Resource Identifier (URI): Generic Syntax. at <http://www.ietf.org/rfc/rfc3986.txt> [accessed 3 June 2014]
- Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A. and D. Reynolds (2013) *RIF Core Dialect (Second Edition)*. W3C. at <http://www.w3.org/TR/rif-core/> [accessed 11 February 2014]
- Brickley, D. (2003) *Basic Geo (WGS84 lat/long) Vocabulary*. W3C. at <http://www.w3.org/2003/01/geo/> [accessed 11 February 2014]
- CEN/TC287 (2012) TR 15449-4:2012 *Geographic information – Spatial Data Infrastructures – Part 4: Service Centric View*.
- Ciardi, G., Abrescia, A. and S. Pezzi (2013). *Linked open data from OGC® compliant web services: the case of Regione Emilia-Romagna GeoPortal*. at http://inspire.jrc.ec.europa.eu/events/conferences/inspire_2013/schedule/submissions/147.doc [accessed 11 february 2014].
- Davidson, P. (2009) Designing URI Sets for the UK Public Sector, UK Chief Technology Officer Council, 2009, Designing URI Sets for the UK Public Sector, at https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/60975/designing-URI-sets-uk-public-sector.pdf [accessed 11 February 2014]
- European Parliament, Council of the European Union (2007). 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).
- Cox, S. (2013). An explicit OWL representation of ISO/OGC Observations and Measurements. *Proceedings of the 6th International Workshop on Semantic Sensor Networks, October 22, 2013, Sydney, Australia*, pp: 1-18.

- López-Pellicer, F.J, Vilches-Blázquez, L.M., Zarazaga-Soria, F.J. Muro-Medrano, P.R., O. Corcho (2012). The Delft Report: Linked Data and the challenges for geographic information standardization. *Revista Catalana de Geografia* vol. XVII, nº 44.
- GeoKnow (2013) Project description at: <http://geoknow.eu/Project.html> [accessed November 2013].
- Geonovum (2012). Informatiemodel Ruimtelijke Ordening 2012. at <http://www.geonovum.nl/wegwijzer/standaarden/informatiemodel-ruimtelijke-ordening-imro2012> [accessed 3 June 2014]
- Hobona, G. and R. Brackin (2011) *OGC® OWS-8 Cross Community Interoperability (CCI) Semantic Mediation Engineering Report*. at https://portal.opengeospatial.org/files/?artifact_id=46342 [accessed 11 February 2014]
- ISA (2012a). ISA Programme Core Vocabularies Specification. https://joinup.ec.europa.eu/system/files/project/Core_Vocabularies-Business_Location_Person-Specification-v1.00_0.pdf [accessed 11 February 2014]
- ISA (2012b) D7.1.3 - Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC, https://joinup.ec.europa.eu/sites/default/files/D7.1.3%20-%20Study%20on%20persistent%20URIs_0.pdf.
- ISO TC/211 - ISO 19107:2003. *Geographic information - Spatial schema*.
- ISO TC/211 - ISO 19109:2005. *Geographic information -- Rules for application schema*.
- ISO TC/211 - ISO 19125-1:2004. *Geographic information - Simple feature access - Part 1: Common architecture*.
- ISO TC/211 - ISO/CD 19150-2 (2013). *Geographic information - Ontology - Part 2: Rules for developing ontologies in the Web Ontology Language (OWL)*. (Draft International Standard)
- ISO/IEC 13249-3 (2011) *Information technology — Database languages — SQL multimedia and application packages — Part 3: Spatial*.
- Janowicz, K., Hitzler, P., Admas, B., Kolas, D., and Vardeman II, C. (2014) Five Stars of Linked Data Vocabulary Use. *Semantic Web Journal* (2014) 1-0.
- Kay, M. (2007) *XSL Transformations (XSLT) Version 2.0*. W3C. at <http://www.w3.org/TR/xslt20/> [accessed 11 February 2014]

- Klyne, G. and J.J. Carroll (2004) *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C. at <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/> [accessed 11 February 2014]
- Lake, R. and A. Cuthbert (2000) *Geography Markup Language (GML) v1.0*. OGC. At http://portal.opengeospatial.org/files/?artifact_id=7197 [accessed 11 February 2014]
- Manola, F. and E. Miller (2004) *RDF Primer*. W3C. At <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> [accessed 11 February 2014]
- Object Management Group (2009). *Ontology Definition Metamodel (version 1.0)*. At <http://www.omg.org/spec/ODM/1.0/> [accessed 11 February 2014]
- OEG (2014), Ontology Engineer Group, *geometr2rdf*, at <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/technologies/151-geometry2rdf> [accessed 3 June 2014]
- Overbeek, H. and L. van den Brink (2013). Towards a national URI--Strategy for Linked Data of the Dutch public sector. At http://www.pilod.nl/wiki/Bestand:D1-2013-09-19_Towards_a_NL_URI_Strategy.pdf [accessed 3 June 2014]
- Overbeek, H. and T. Brentjens (2013). Draft URI Strategy for the NL Public Sector. At http://www.w3.org/2013/04/odw/odw13_submission_14.pdf [accessed 3 June 2014]
- Smits, P., Arctur, D, Atkinson, R., Bargmeyer, B., Boerboom, L., Browdy, S. F., Praag, E. V. , Hodge G., Jensen, S., Ulgen, T. S. and M. Wilson (2011) "Recommendations for the technical design of a global interoperable information network," *Eye on Earth Working Group 3 – Technical Infrastructure, White Paper 1 EOEWG3/1*. At <http://www.eyeonearthsummit.org/working-groups/working-group-3> [accessed 11 February 2014].
- Perego, A., Lutz, M. and P. Archer, (2013). *ISA Programme Location Core Vocabulary*. W3C. at <http://www.w3.org/ns/locn> [accessed 11 February 2014]
- Perry, M. and J. Herring (2010) *OGC GeoSPARQL - A Geographic Query Language for RDF Data*. At <http://www.opengeospatial.org/standards/geosparql> [accessed 11 February 2014]
- Portele, C. (2012) OGC® *Geography Markup Language (GML) — Extended schemas and encoding rules*. At https://portal.opengeospatial.org/files/?artifact_id=46568 [accessed 11 February 2014]

- Schade, S. and S. Cox (2010) Linked Data in SDI or How GML is not about Trees. *Proceedings of the 13th AGILE International Conference on Geographic Information Science, 10-14 May 2010, Guimarães, Portugal*
- Stadler, C., Lehmann, J., Höffner, K., and S. Auer, (2012) LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web Journal*, 3(4):333-354.
- Tschirner, S, D. Scherp, and S. Staab (2011), Semantic Access to INSPIRE - How to Publish and Query Advanced GML Data, *Proceedings of the Terra Cognita Workshop on Foundations, Technologies and Applications of the Geospatial Web In conjunction with the International Semantic Web Conference (ISWC2011), 23 October 2011, Bonn, Germany.*
- W3C OWL Working group (2012). OWL 2 Web Ontology Language Document Overview (Second Edition). W3C. At <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/> [accessed 3 June 2014]
- W3C (2012). "223 Best Practices URI Construction", at http://www.w3.org/2011/gld/wiki/223_Best_Practices_URI_Construction [accessed 3 June 2014]
- LOGD (2013). Linking Open Government Data, URI Design Principles: Creating Persistent URIs for Government Linked Data, Tetherless World Constellation(TWC) at Rensselaer Polytechnic Institute (RPI). At <http://logd.tw.rpi.edu/instance-hub-uri-design> [accessed 3 June 2014]