International Journal of
*Geo-Information*

MDPI

*Article*

# Adapted Rules for UML Modelling of Geospatial Information for Model-Driven Implementation as OWL Ontologies

**Knut Jetlund [1,2,*], Erling Onstein [2] and Lizhen Huang [2]**

[1] Norwegian Public Roads Administration, 2318 Hamar, Norway
[2] Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology, 2821 Gjøvik, Norway
[*] Correspondence: knut.jetlund@vegvesen.no; Tel.: +47-9596-2863

Received: 28 June 2019; Accepted: 19 August 2019; Published: 22 August 2019

**Abstract:** This study aims to improve the implementation of models of geospatial information in Web Ontology Language (OWL). Large amounts of geospatial information are maintained in Geographic Information Systems (GIS) based on models according to the Unified Modeling Language (UML) and standards from ISO/TC 211 and the Open Geospatial Consortium (OGC). Sharing models and geospatial information in the Semantic Web will increase the usability and value of models and information, as well as enable linking with spatial and non-spatial information from other domains. Methods for conversion from UML to OWL for basic concepts used in models of geospatial information have been studied and evaluated. Primary conversion challenges have been identified with specific attention to whether adapted rules for UML modelling could contribute to improved conversions. Results indicated that restrictions related to abstract classes, unions, compositions and code lists in UML are challenging in the Open World Assumption (OWA) on which OWL is based. Two conversion challenges are addressed by adding more semantics to UML models: global properties and reuse of external concepts. The proposed solution is formalized in a UML profile supported by rules and recommendations and demonstrated with a UML model based on the Intelligent Transport Systems (ITS) standard ISO 14825 Geographic Data Files (GDF). The scope of the resulting ontology will determine to what degree the restrictions shall be maintained in OWL, and different conversion methods are needed for different scopes.

**Keywords:** geographic information systems; information model; model driven architecture; unified modelling language; web ontology language

## 1. Introduction

### 1.1. Geospatial Information

This article presents novel research continued from the study presented in [1].

Geospatial information is a vital part of the knowledge about real-world features and events. Sharing and reusing geospatial information and non-spatial information from heterogeneous sources is of paramount importance for domains such as Smart Cities [2]; disaster management [3,4]; construction and asset management [5–7]; and Life Cycle Assessment (LCA) [8]. One specific example is the domain of Intelligent Transport Systems (ITS), where sensors in vehicles and road-side equipment collect and share vast amounts of data about weather conditions, traffic events, regulations and road environments. The collected data depends on location references to become valuable information and needs to be combined with geospatial information from other sources in order to build the knowledge needed for legal and safe navigation [9].

Standards and specifications developed by the International Standardization Organization, Technical Committee 211 (ISO/TC 211) and the Open Geospatial Consortium (OGC) provide the foundation for structured geospatial information. Local, national and regional authorities, agencies and organizations worldwide have applied the standards for collecting and maintaining large amounts of structured geospatial information covering a wide range of purposes. Information models, databases and applications for geospatial information are based on standards from ISO/TC 211 and OGC [9,10].

Modelling of geospatial information according to ISO TC/211 and OGC standards is a process where a portion of the real world—known as a universe of discourse—is perceived in a context of geographic application and defined in a conceptual model. ISO/TC 211 uses the term "geographic information", while OGC uses the term "geospatial information". The terms "geographic" and "geospatial" are considered equivalent in this article. The conceptual model is represented in a conceptual schema, formalized by the Unified Modeling Language (UML) [11]. Figure 1 illustrates the process.

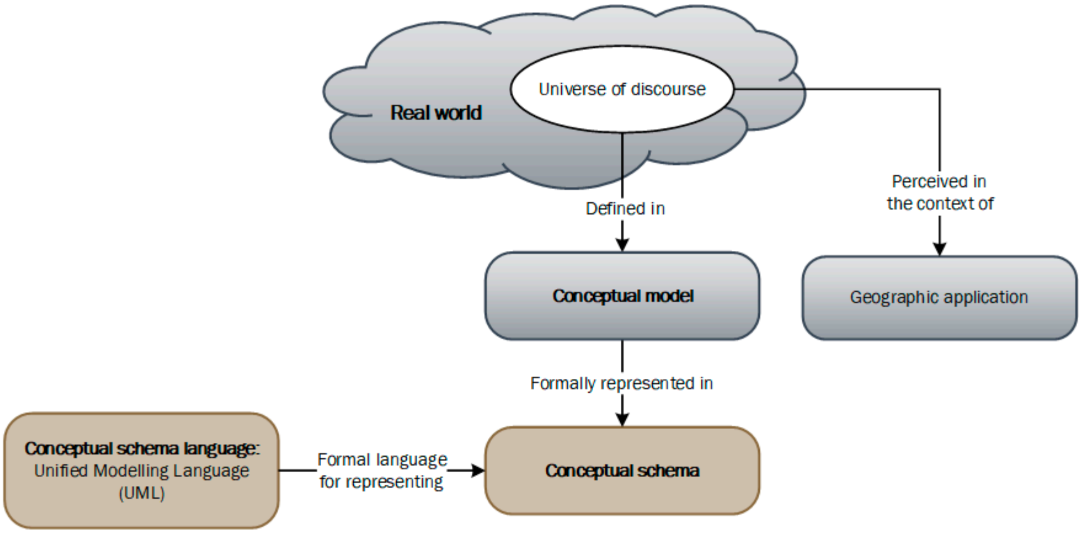

Real world
Universe of discourse
Defined in
Perceived in the context of
Conceptual model
Geographic application
Formally represented in
Conceptual schema language: Unified Modelling Language (UML)
Formal language for representing
Conceptual schema

**Figure 1.** The ISO/TC 211 process of information modelling. Adapted from [12,13].

According to Model Driven Architecture (MDA) [14], the conceptual schemas shall be independent of any implementation technology but can be converted to implementation schemas in database and exchange formats. The core standards ISO 19103 [13] and ISO 19109 [12] define the ISO/TC 211 UML profile, the General Feature Model (GFM) for geospatial information and rules for semantics needed for conversion to implementation schemas. ISO 19136 [15] defines specific UML modelling rules and conversion rules for implementation in the exchange format GML. Figure 2 illustrates the four levels of abstraction in MDA according to ISO 19103: metamodels; abstract conceptual schemas; conceptual application schemas; and implementation schemas. Furthermore, the figure illustrates how conceptual schemas are converted to implementation schemas in the two formats GML and OWL.

Structured geospatial information based on ISO/TC 211 and OGC standards has mainly been stored and maintained in relational databases with extensions for geometry, topology and geospatial operations, and has been accessed through specialized GIS software [16]. Over the last 10 to 15 years, service standards such as Web Map Services (WMS) [17] and Web Feature Services (WFS) [18] have been used extensively to share geospatial information and information models on the World Wide Web. Spatial Data Infrastructures (SDI) provide portals and catalogue services for searching and accessing information from different service providers. However, even though WMS and WFS are based on general IT standards, they are specific standards for geospatial information, with limited possibilities for linking the information with other sources on the World Wide Web [19,20].

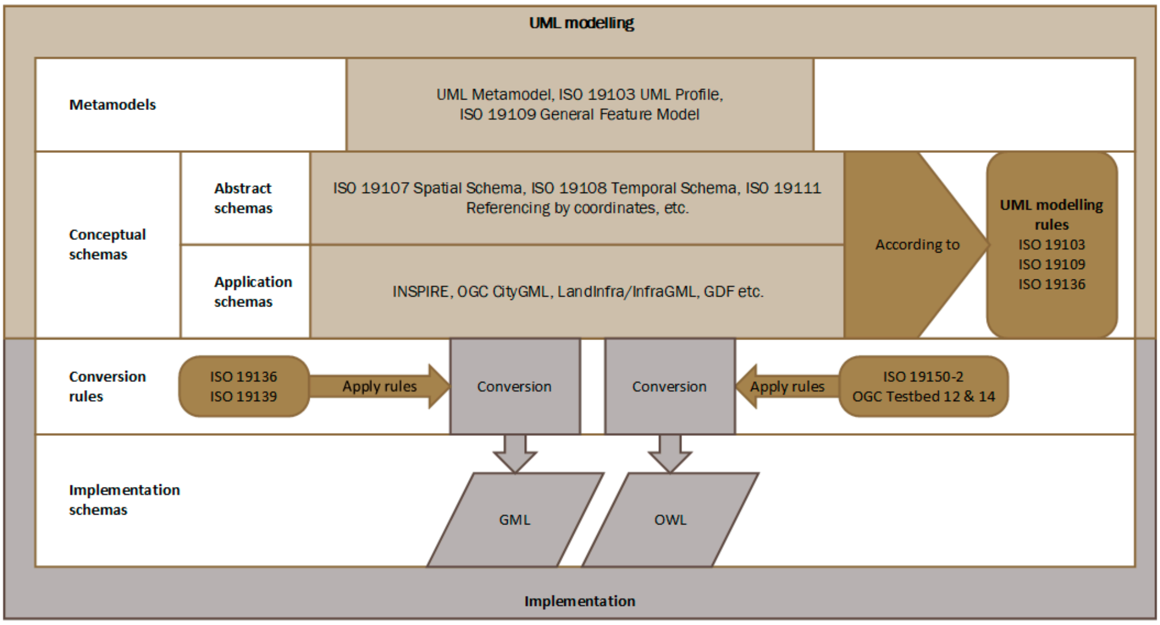


**Figure 2.** UML modelling and conversion to implementation schemas according to ISO/TC 211 and OGC standards.

### *1.2. Geospatial Information in the Semantic Web*

The Semantic Web provides the concepts for structured and machine-readable descriptions of any kind of information on the World Wide Web, independent of applications and domains. Information is described in the Resource Description Framework (RDF) [21] as graphs of triples with subject, predicate and object. Information models are described as ontologies in the Web Ontology Language (OWL) [22], which is based on RDF.

Enabling information models and geospatial information based on ISO/TC 211 and OGC standards in RDF and OWL can bring more value to the domain of GIS and other domains as well as the Semantic Web in general [23]. Semantic Web technologies can be used to improve discovery in SDIs [24–29]. Authoritative geospatial information can be combined with less structured geospatial information from crowdsourcing, citizen science and sensors [20,30], and Semantic Web concepts for reasoning may be used to derive new knowledge [31–33]. Most importantly, geospatial information can be easier accessed from outside the GIS domain, linked with non-spatial information and reused between stakeholders and domains.

The availability of standardized geospatial information on the Semantic Web is still limited, but some effort has been put into the issue by OGC and W3C, who established The Spatial Data on the Web Working Group in cooperation in 2015 [34]. ISO/TC 211 developed the standard ISO 19150, part 2 [35] with rules for conversion from UML to OWL. The ISO/TC 211 Group for Ontology Management (GOM) [36] derived OWL ontologies for most ISO/TC 211 standards following the rules from ISO 19150-2. Further work has been done in INSPIRE [37] and OGC [38,39]. The Ordnance Survey in Ireland has published more than 50,000,000 spatial objects in their Prime2 database [40–42]. Mapping authorities in several other countries have published data sets or researched the subject as well, including, but not limited to; the United Kingdom [43], The Netherlands [44,45], Turkey [31], Spain [46] and Greece [47].

Some of the OWL ontologies for the published data sets have been converted from UML models, while others have been developed directly in OWL, parallel to existing UML models.

### *1.3. Contribution and Research Questions*

This study investigates conversions from conceptual UML models according to ISO/TC 211 and OGC standards to OWL ontologies, as illustrated in Figure 2. The study aims to improve the process and achieve more precise implementation in OWL. Two research questions have been studied:

1. What are the primary challenges in conversions from UML models of geospatial information to OWL ontologies?
2. How can conversion challenges be overcome with adapted rules for UML modelling?

The scope of the study is limited to information models and conversion from UML to OWL. Challenges related to managing geospatial information in the Semantic Web, such as spatial indexing, complex geometries and operations are considered out of scope.

## **2. Materials and Methods**

### *2.1. Literature Search*

The primary purpose of the literature search was to identify relevant literature for answering the research questions. The results include literature where UML and OWL were compared and literature that described conversions from UML models to OWL ontologies. Besides, literature that described use cases and experiences with geospatial information in the Semantic Web was considered relevant for a broader view on the issue.

Combinations of keyword sets, listed in Table 1, were used in the search portals Oria, Web of Science and Google Scholar. Combinations of sets 1–3 were used in the first search for literature about geospatial UML models and OWL ontologies, preferably in the title: "1 and 2"; "2 and 3"; and "1 and 2 and 3", while sets 4 and 5 were used for refining the searches. Finally, keyword set 6 was used in combination with keyword sets 3 and 4 to find literature about structured geospatial information on the Semantic Web.

**Table 1.** Keyword sets for the literature search.

| | Keyword Set | Purpose—Literature Mentioning |
|---|---|---|
| 1. | "UML" | The abbreviation UML |
| 2. | "OWL" OR "Ontology" OR "Ontologies" | The abbreviation OWL or ontologies in general. |
| 3. | "Geospatial" OR "Geographic" OR "Spatial" OR "GIS" | Terms for geospatial information. |
| 4. | ("ISO AND "211") OR "OGC" | The geospatial standardization actors ISO/TC 211 and OGC. |
| 5. | "Mapping" OR "Conversion" OR "Transformation" | Terms for conversion processes. |
| 6. | "Semantic Web" OR "Linked Data" | Terms for the Semantic Web and Linked Data. |

The initial search results were refined by studying titles and abstracts, while the final selection was made by studying the content. Inspection of cited literature (backward search) and literature that cited the selected literature (forward search) identified additional relevant literature. Besides, relevant standards, specifications and reports were found through searches in standards catalogues and on the World Wide Web. The final results from the searches are presented in Table 2.

**Table 2.** Selected literature per subject and literature type.

| Subject | Literature Type | | | | | |
|---|---|---|---|---|---|---|
| | Book or Book Section | Conference Paper or Proceedings | Journal Article | Report or Standard | Web Page | Total |
| Geospatial information in the Semantic Web | 4 | 8 | 34 | 4 | 3 | 53 |
| Comparing UML and OWL | 10 | 2 | 8 | 1 | 1 | 22 |
| UML to OWL conversions in general | 1 | 4 | 9 | 0 | 0 | 14 |
| UML to OWL conversions for geospatial information | 1 | 5 | 6 | 2 | 1 | 15 |

Results from the literature search on geospatial information in the Semantic Web are the foundation for the overall description in the introduction of this article. Results from the searches on UML and OWL are further discussed in the subsequent state-of-the-art study.

## 2.2. State of the Art

### 2.2.1. Comparing UML and OWL

Fundamental differences that are essential for conversions between UML and OWL were pointed out in our state-of-the-art study presented in [1]. One fundamental difference in modelling approaches is the assumption of an open or closed world. OWL and the Semantic Web is founded on an Open World Assumption (OWA) and the possibility for anyone to say anything about anything (AAA). An information model following the OWA describes the real world only as it is known at the time, and more information which is outside of the current knowledge may be added. No conclusions that are assuming all information is available can be drawn, and nothing is true or false unless explicitly stated [48]. UML is following a Closed World Assumption (CWA), where the information model is assumed to be a complete description of the real world in a given context (e.g., specific use of geographic information). A feature type in an ISO/TC 211 compliant UML model is a classification of real-world phenomena with common characteristics, perceived in the context of a geographic application. A feature instance in a data set shall be of one single feature type, with characteristics as single property types, as illustrated in Figure 3. Contrary, an individual instance in an open world of OWL can be linked to several classes and have a flexible set of properties [23]. Because of these different assumptions, UML models have implicit restrictions that may need to be specified if they shall be maintained in OWL.

Another key principle in the Semantic Web is linking, reusing and extending concepts from other ontologies—including ontologies from other domains. UML models are defined in a closed environment with less flexibility for linking to external concepts. UML models developed according to ISO/TC 211 standards are reusing concepts from other models, but the reuse is mainly limited to models from the domain of geospatial information.

One fundamental logical difference is that OWL is based on set theory and description logic, with set-based class constructors such as union, intersect and complement (e.g., an OWL class may be constructed as the union of two other classes, meaning that it contains all instances of the two classes). UML does not have set-based class constructors, but some of the implicit restrictions in UML models must be translated to class constructors in OWL. Furthermore, OWL classes and properties are individuals which may be queried in the same manner as, and in combination with, the individuals in the data set. A UML model must be realized as an implementation schema (e.g., XML or a database schema), and there is a clear distinction between the schema and the instances in the data set. Finally, the concepts for class properties are different in the two technologies: while each class in UML is the single owner of its attributes and associations, properties are individual concepts and globally defined in OWL, and may be assigned to any class.

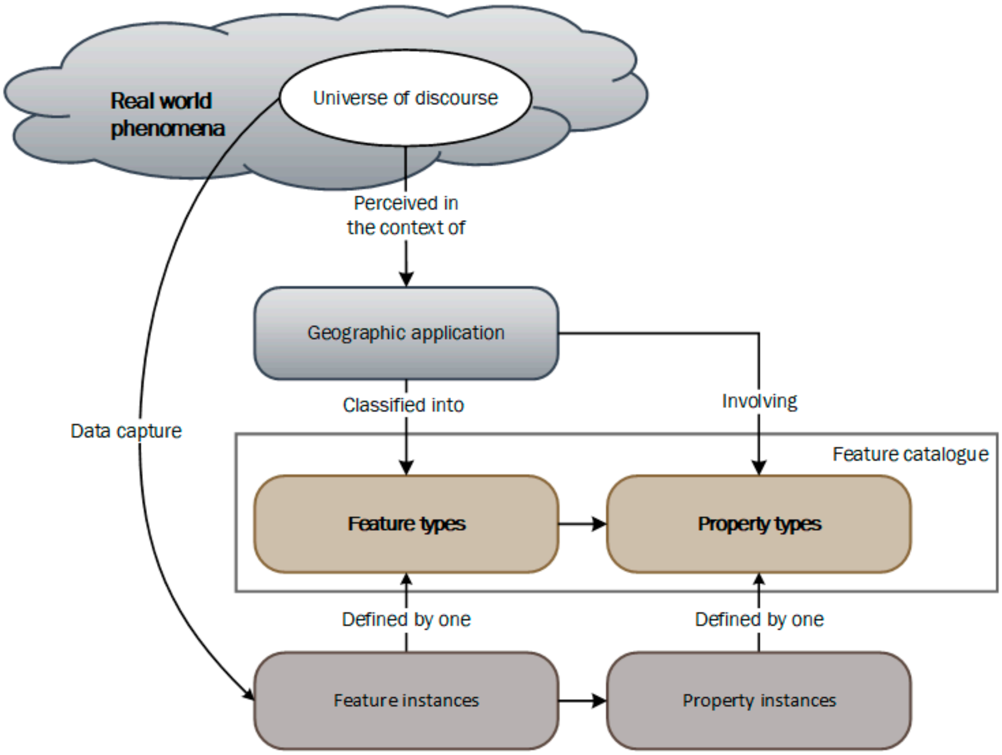


**Figure 3.** From the real world to data. Adapted from [12].

Despite the differences between UML and OWL, some effort has been put in developing UML profiles for applying UML as a tool for OWL ontology development [49–61]. The motivation has been to exploit the standardized graphical notation in UML—which is very useful for human communication. Some applications for ontology development include a graphical presentation of the ontologies, but there is no standardized graphical view equivalent to UML class diagrams. Most prominent among the UML profiles for OWL is the Ontology Definition Metamodel (ODM) [60], developed by the Object Management Group (OMG). The ODM contains a formal metamodel and UML profile for RDF and OWL.

A study that compared modelling of geospatial information in UML based on ISO/TC 211 MDA and OWL Ontologies was described in [31]. The UML-based model was considered the most robust and prepared model for use and linking with other datasets within SDIs, whereas the OWL-based model provided better support for sharing, discovery and linking with any other information on the Semantic Web. The authors of [31] suggested that both approaches should be used, but did not discuss whether models based on one approach should be the original and the other in a derived model, or whether the models should be developed and maintained in parallel.

### 2.2.2. Conversions from UML to OWL

The state-of-the-art study of conversions from geospatial UML models to OWL ontologies presented in [1] was extended for this article with work described in additional sources and more detailed analysis of conversion rules. The main source of the studies is the conversion rules defined in ISO 19150, part 2 [35]. The AR3NA project refined and modified the rules from ISO 19150-2 into guidelines for RDF encoding of geospatial information and models defined according to the INSPIRE Directive in Europe [37]. The OGC Testbeds 12 [38] and 14 [39] focused on specific geospatial UML to OWL issues and particularly on the implementation of rules in the conversion software ShapeChange [62]. Finally, several studies described conversions between UML and OWL in general [63–69] while other described specific conversions of UML models of geospatial information to OWL [70–72].

Table 3 summarizes rules for conversion from UML to OWL for fundamental concepts used in UML models for geospatial information. Further details are discussed in [1] and in subsequent sections.

**Table 3.** Summary of conversion rules.

| UML Concept | OWL Concept | Conversion Rule Specification |
|---|---|---|
| Package | Ontology | Name and structure as in UML [35].<br>Name and structure from tagged values [37,38]. |
| Class | Class | Direct conversion.<br>Subclass of AnyFeature [35].<br>Mapping to external classes [38]. |
| Class generalization | subclassOf | Direct conversion. |
| Abstract class | Not existing | isAbstract annotation property [35,37,38].<br>DisjointUnion axiom [72]. |
| Primitive data type | DatatypeProperty | Matched to XSD Datatypes. |
| Structured data type | DatatypeProperty or ObjectProperty | Mapping to a few external types, else new class [35].<br>Mapping to specified external types, else new class [38].<br>Mapping to all similar external types, else new class [37]. |
| Spatial data types | Data types defined in ISO 19107 [73] and GeoSPARQL [74] | Data types defined in the ISO 19107 ontology [35].<br>Mapping to GeoSPARQL data types [3,28,29,34,37,44].<br>Extending GeoSPARQL [31,41,42]. |
| Enumeration | DataOneOf | Direct conversion.<br>SKOS Concept Scheme [37,38]. |
| Code lists | Several options | SKOS and allValuesFrom [35].<br>SKOS [37,38].<br>DataUnionOf and any value [72]. |
| Union | Several options | Union [35].<br>Intersections and Union [37,38].<br>Flattening [38]<br>Subproperty and ExactCardinality [72].<br>DisjointClasses [75]. |
| Attribute and association role | Property | Simple conversion [68,69].<br>Globalization by similarity matching [71].<br>Globalization by prefix [35,38].<br>Global attributes with domain AnyFeature [35].<br>Globalization by manual matching [37,38].<br>Mapping to external properties [38,67]. |
| Simple association | Domain and range | Direct conversion. |
| Aggregation | Not existing | Hierarchy of properties [66].<br>aggregationType annotation property [35]. |
| Composition | Not existing | Hierarchy of properties [66].<br>InverseFunctional [67].<br>aggregationType annotation property [35]. |

### 2.2.3. Packages

The package concept in UML corresponds to an OWL ontology. However, packages in UML may also be used for an informal structure that does not need to be reflected in the implementation. ShapeChange can use tagged values on the packages to define the output structure, following recommendations from OGC Testbed 12.

### 2.2.4. Classes

The class concept exists with similar semantics in UML and OWL—including generalization and inheritance between classes. ISO 19150-2 specifies that classes defined as feature types according to ISO 19109 shall be declared as subclasses of the class AnyFeature from the ISO 19109 ontology.

OGC Testbed 12 discussed mapping from classes in UML models to equivalent classes already defined in OWL ontologies. The mapping was implemented in ShapeChange and can be defined in a configuration file.

Abstract classes in UML are used in generalizations, in which the abstract class is a superclass whose purpose is to define attributes, associations and operations that are common to all subclasses. The abstract class shall not have instances in an implementation, which is a type of restriction that does not exist in OWL. As discussed in [1], none of the identified conversion rules maintains the implicit restriction of the abstract class concept.

### 2.2.5. Data Types

Data types for attributes in UML may be equal or similar in models from different domains (e.g., an integer value is an integer value regardless of whether the model describes geospatial information or any other kind of information). As discussed in [1], ISO 19150-2 defines a mapping from a set of data types to XML Schema types. The INSPIRE Guidelines emphasize more reuse of data types from existing ontologies with an extended list of predefined mappings of commonly used data types. ShapeChange has implemented a configurable mapping that can be set individually for each model.

### 2.2.6. Spatial Data Types

The standard ISO 19107 [73] defines specific data types for geometry and topology, while ISO 19125-1 [76] defines a simplified profile of ISO 19107. Data types from ISO 19107 and ISO 19125-1 are reused in other ISO/TC 211 and OGC standards as well as other models based on the standards. ISO 19150-2 specifies that the ISO 19107 ontology—which has been derived from the ISO 19107 UML model—shall be used for spatial attributes in application schemas. A list of valid spatial object types is provided in the standard.

The OGC GeoSPARQL [74] specification includes a vocabulary for describing geometry in RDF and OWL. The INSPIRE Guidelines map data types from ISO 19107 to the generic GeoSPARQL Geometry class or one of its subclasses, which are ontology representations of the data types from either ISO 19125-1 or ISO 19136 (GML). This approach was used in several studies and projects for the publication of geospatial information in the Semantic Web as well. ShapeChange is configurable and can support several approaches. For practical use, there should not be any difference between using data types from GeoSPARQL or the ISO 19107 ontology, as they are initially based on the same conceptual model from ISO 19107.

### 2.2.7. Enumerations

ISO 19150-2 defines the conversion of UML enumerations to the "OneOf" axiom with a collection of values from the UML enumeration. OGC Testbed 12 discussed an alternative conversion where enumerations were treated as code lists. Both approaches have been implemented in ShapeChange. The INSPIRE Guidelines follow the rule from ISO 19150-2 for enumerations with self-describing codes that have a distinct meaning. In other cases, the enumeration shall be handled as code lists.

### 2.2.8. Code Lists

The code lists concept is explicitly defined in ISO 19103. Code lists are flexible enumerations, meaning that there can be other values than those described in the list. ISO 19150-2 defines the code list in OWL as three parts based on SKOS concept schemes. OGC Testbed 12 discussed several methods for handling code lists in OWL and focused on the management of the lists as external references outside of the ontology. The INSPIRE Guidelines follow the approach from OGC Testbed 12 with separate external SKOS concept schemes for the code lists. The ontologies include the URL to the SKOS Concept Scheme, but without any formal binding.

An alternative approach was described in [72], with a union of a "OneOf" list with the code list values and any other value, defined with a standard XML Schema expression.

### 2.2.9. Unions

The union concept is explicitly defined in ISO 19103 and is different from the set-based union class constructor in OWL. A union in UML models according to ISO 19103 is a list of several alternative datatypes, where one and only one shall be used for an attribute value. ISO 19150-2 defines the conversion of a UML union directly to an OWL union of classes. OGC Testbed 12 pointed out several weaknesses with this conversion rule and described a different approach that is also used in the INSPIRE Guidelines. They used a combination of intersections where only one member could have a value, combined in a union that combined all the intersections. Two other approaches where the union was flattened were described as well. They believed the flattened approaches would work better in Semantic Web software.

### 2.2.10. Attributes and Association Roles

UML classes can have two kinds of properties: attributes or associations to other classes. Both kinds of properties are equivalent to OWL properties and are handled as such in the studied conversion rules. However, as discussed in [1], one fundamental difference requires special treatment: properties in OWL are individual concepts that are globally scoped, while UML properties are unique within each class. Several classes in a UML model may have identical or almost identical properties, or they may have properties with an identical name, but different data type or definition. Identical UML properties should be handled as one global OWL property, while UML properties with identical names but different data type or definition need to be handled as separate OWL properties.

The core rule for globalization of UML properties in ISO 19150-2 is to add the class name as a prefix to the property name, which makes all properties locally defined for each class, but globally unique. Reused attributes can alternatively be defined without prefix, and it is suggested to use the generic class "AnyFeature" as the domain. The INSPIRE Guidelines supplies the rules with more reuse of existing properties from external ontologies. UML properties with identical or close to identical meaning are converted to OWL properties with a global scope, and UML properties that can be matched to properties already defined in other ontologies are converted to those properties in OWL. Similarity matching, as described in [71], may be used to reduce the manual work involved in the process of identifying potential global and external properties.

The issue of property globalization was discussed in OGC Testbed 12 as well, with four approaches that were implemented in ShapeChange: all properties locally defined with class name prefix as in 19150-2; all global properties defined in one UML class; global properties defined in a configuration file; or global properties identified by globally unique names. Besides, ShapeChange can be configured to map UML properties to externally defined OWL properties, as discussed in [44].

### 2.2.11. Associations

Aggregations and composite associations add more semantics to UML models. An UML aggregation defines one class as a part of a main class (the whole) but does not add any restrictions to the model. The aggregation merely informs about a closer relationship. A UML composition defines a more restricted relationship between two classes. An instance of a part class in a composition can only be related to one whole instance. The restriction defined by a composition does not exist directly in OWL, and, as discussed in [1], none of the identified conversion rules maintains the restriction.

## 3. Results

### *3.1. Semantics in UML Models for Implementation as OWL Ontologies*

Model-driven conversion from conceptual UML models to implementation schemas as illustrated in Figure 2 relies on a specific use of concepts and inclusion of specific semantics in the UML models. Rules defined in ISO 19103, ISO 19109 and ISO 19136 state how general UML concepts shall be used for models of geospatial information, supplemented with specialized UML stereotypes and tagged values for additional semantics. The standards ISO 19103 and ISO 19109 defines general rules for UML models of geospatial information, while ISO 19136 describes specific rules for UML models that shall be implemented as GML schemas. However, specific rules for semantics that could support conversion to OWL ontologies are not defined in any of the identified standards.

Most of the studies described in the state-of-the-art study focused on rules for conversion from UML models to OWL ontologies. Less attention was put on how to develop UML models that are prepared for implementation in OWL. The issue was briefly discussed in [28], where they suggested that conversion challenges could be overcome by improving the original UML models. Semantics for linking UML attributes to externally defined OWL properties were suggested in [44], while OGC Testbed 12 discussed and implemented conversion rules in ShapeChange, based on specific tagged values. The UML profiles for OWL described in [49–61] specified how UML could be used to develop RDF and OWL ontologies specifically. However, the profiles differ significantly from the profile in ISO 19103 and cannot be applied to models according to ISO/TC 211 standards. Furthermore, the MDA approach in ISO/TC 211 standards is to develop conceptual models that can be implemented in several formats, not models scoped for one single implementation format.

Two of the core challenges for the conversion from a closed UML world to an open OWL world can potentially be overcome by adding more semantics to UML models: defining attributes in UML for implementation as global properties in OWL, and linking internal UML classes, data types and properties to existing external OWL concepts.

### *3.2. Extended UML Profile for Geospatial Information*

Additional semantics in UML models are handled by profiles that extend concepts in the UML metamodel. The primary construct to be used in profiles is the stereotype. Stereotypes extend existing metaclasses and can specify semantics as properties for UML concepts, referred to as tagged values [11]. The formal UML profile for geospatial information is defined in ISO 19103. ISO 19109 and ISO 19136 describe additional semantics as well—but not as formal profiles. Furthermore, the INSPIRE Generic Conceptual Model [77] describes specific tagged values for INSPIRE information models. Finally, the conversion rules developed in OGC Testbed 12 and implemented in ShapeChange use some tagged values not specified in standards.

We suggest an extended UML profile for geospatial information that includes semantics needed for conversion to OWL. The profile is based on ISO 19103, ISO 19109 and ISO 19136, and includes extensions from INSPIRE and ShapeChange. Figure 4 illustrates the extended UML profile, while the tagged values that are added in the profile are described in Table 4. Subsequent sections describe how the extended semantics can be utilized for improved conversion from UML to OWL.

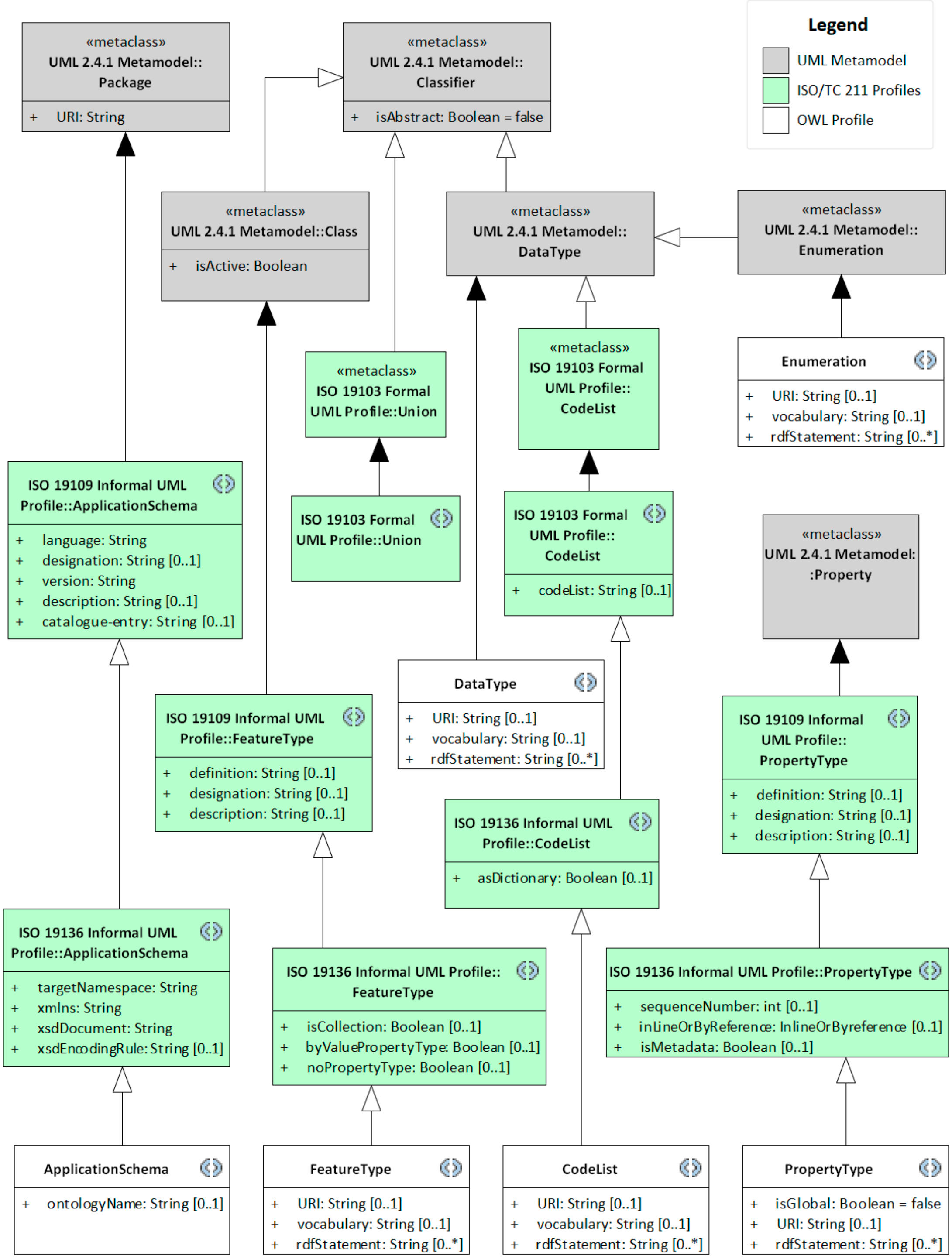

Legend
UML Metamodel
ISO/TC 211 Profiles
OWL Profile
«metaclass» UML 2.4.1 Metamodel:: Package
+ URI: String
«metaclass» UML 2.4.1 Metamodel:: Classifier
+ isAbstract: Boolean = false
«metaclass» UML 2.4.1 Metamodel::Class
+ isActive: Boolean
«metaclass» UML 2.4.1 Metamodel:: DataType
«metaclass» UML 2.4.1 Metamodel:: Enumeration
«metaclass» ISO 19103 Formal UML Profile::Union
«metaclass» ISO 19103 Formal UML Profile:: CodeList
Enumeration
+ URI: String [0..1]
+ vocabulary: String [0..1]
+ rdfStatement: String [0..*]
ISO 19109 Informal UML Profile::ApplicationSchema
+ language: String
+ designation: String [0..1]
+ version: String
+ description: String [0..1]
+ catalogue-entry: String [0..1]
ISO 19103 Formal UML Profile::Union
ISO 19103 Formal UML Profile:: CodeList
+ codeList: String [0..1]
«metaclass» UML 2.4.1 Metamodel: :Property
DataType
+ URI: String [0..1]
+ vocabulary: String [0..1]
+ rdfStatement: String [0..*]
ISO 19109 Informal UML Profile::FeatureType
+ definition: String [0..1]
+ designation: String [0..1]
+ description: String [0..1]
ISO 19136 Informal UML Profile::CodeList
+ asDictionary: Boolean [0..1]
ISO 19109 Informal UML Profile:: PropertyType
+ definition: String [0..1]
+ designation: String [0..1]
+ description: String [0..1]
ISO 19136 Informal UML Profile::ApplicationSchema
+ targetNamespace: String
+ xmlns: String
+ xsdDocument: String
+ xsdEncodingRule: String [0..1]
ISO 19136 Informal UML Profile:: FeatureType
+ isCollection: Boolean [0..1]
+ byValuePropertyType: Boolean [0..1]
+ noPropertyType: Boolean [0..1]
ISO 19136 Informal UML Profile::PropertyType
+ sequenceNumber: int [0..1]
+ inlineOrByReference: InlineOrByreference [0..1]
+ isMetadata: Boolean [0..1]
ApplicationSchema
+ ontologyName: String [0..1]
FeatureType
+ URI: String [0..1]
+ vocabulary: String [0..1]
+ rdfStatement: String [0..*]
CodeList
+ URI: String [0..1]
+ vocabulary: String [0..1]
+ rdfStatement: String [0..*]
PropertyType
+ isGlobal: Boolean = false
+ URI: String [0..1]
+ rdfStatement: String [0..*]

**Figure 4.** Extended UML Profile for geospatial information.

**Table 4.** Tagged values in the extended UML profile.

| Tagged Value | Extended Concepts | Description |
|---|---|---|
| ontologyName | Package (ApplicationSchema) | Ontology name, if different from both package name and xsdDocument. |
| URI | Class (FeatureType)<br>DataType<br>Enumeration<br>CodeList<br>PropertyType | URI for the element, for internal and external references. |
| vocabulary | Class (FeatureType)<br>DataType<br>Enumeration<br>CodeList | Reference to an external vocabulary that will replace the internal concept in an OWL implementation. Specified for INSPIRE and implemented in ShapeChange. |
| rdfStatement | Class (FeatureType)<br>DataType<br>Enumeration<br>CodeList<br>PropertyType | One or more RDF statements for linking internal and external concepts. |
| isGlobal | PropertyType | Specifies if the scope of a property is local or global. Default set to false. |

### *3.3. Global Properties in UML*

The rules for converting UML attributes to OWL properties, defined in ISO 19150-2 and OGC Testbed 12, can be used to ensure that OWL properties are globally unique. Each OWL property will either be assigned to one specific class (locally scoped) or any class (globally scoped). However, the rules do not cover the situation where an attribute in UML is reused for several but not all classes—which is a typical situation in UML.

Figure 5 shows examples of reuse in a UML Model based on the ITS standard ISO 14825 Geographic Data Files (GDF) [78]. The model has been modified to become conformant to ISO 19109, and to fit for implementation in OWL. The example contains two abstract superclasses and five implementable classes: The abstract class "RoadFurniture" with subclasses "TrafficSign", "PedestrianCrossing" and "Lighting" and the abstract class "PublicTransportFeature" with subclasses "StopPoint" and "RoutePoint". Two attributes are reused in several classes: The attribute "displayClass" is used in the classes "TrafficSign" and "PedestrianCrossing", while the attribute "accessibility" is used in the two classes "StopPoint" and "PedestrianCrossing". These attributes should be globally unique properties in an OWL implementation, assigned only to the classes they are part of in the UML model.

One approach to avoid duplication of attributes in UML classes is to define superclasses where the reused attributes are defined, with generalization associations from each class to the superclass. However, this is not a viable solution for the example of reuse in Figure 5, as each of the two attributes "displayClass" and "accessibility" shall only be used in two out of five classes. One would need to have multiple combinations of inheritance from one superclass with the "displayClass" attribute and one with the "accessibility" attribute. A larger model with many combinations of reuse would become an intricate spider web of superclasses and generalizations.

OGC Testbed 12 discussed another approach for handling reused attributes, as an issue for future improvement of the conversion rules. They suggested that such attributes could be identified with tagged values in UML and handled in OWL with a union of all classes that have the attribute in the UML model. Our solution follows up the suggestion from OGC Testbed 12 and combines it with one of the approaches that are implemented in ShapeChange, which is the use of a specific class for common attribute concepts. We suggest using a specific and abstract class that contains the original description of global attributes, as illustrated in Figure 6. The class is called "AttributeCatalogue" in the example model and is simply a container for global attributes. The class shall not have any instances in an implementation.

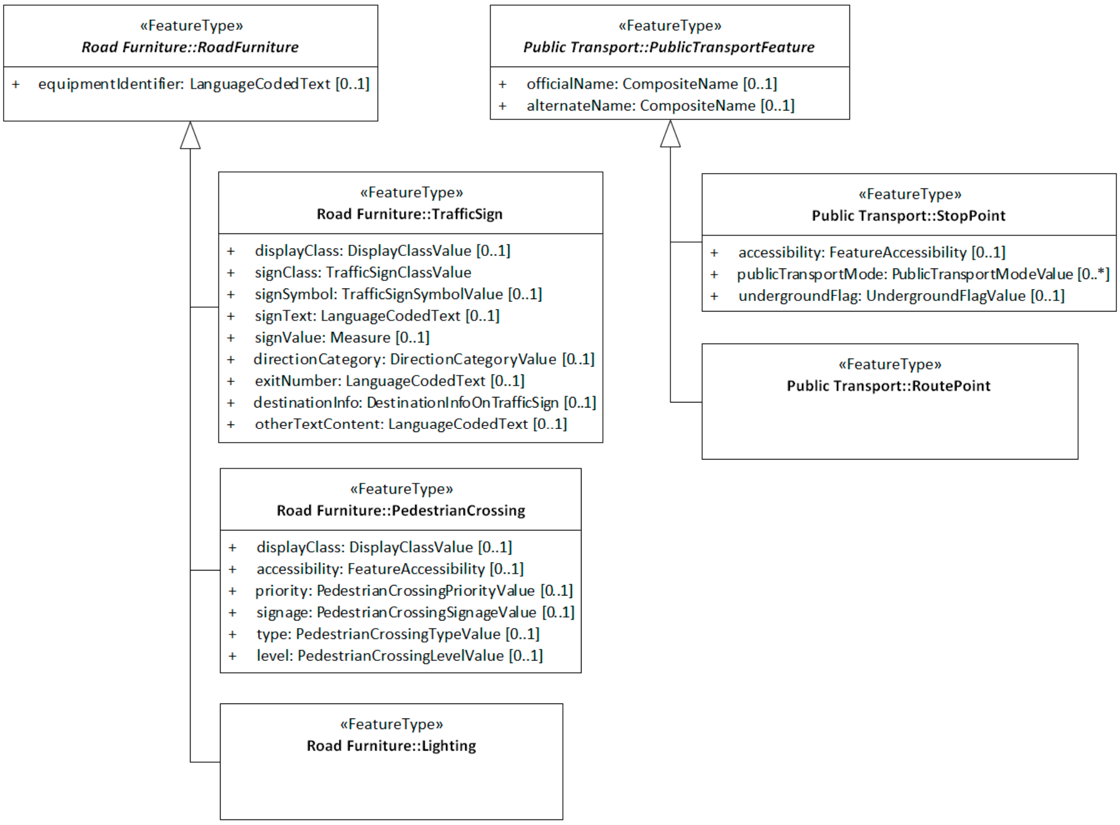

«FeatureType»
Road Furniture::RoadFurniture
+ equipmentIdentifier: LanguageCodedText [0..1]
«FeatureType»
Public Transport::PublicTransportFeature
+ officialName: CompositeName [0..1]
+ alternateName: CompositeName [0..1]
«FeatureType»
Road Furniture::TrafficSign
+ displayClass: DisplayClassValue [0..1]
+ signClass: TrafficSignClassValue
+ signSymbol: TrafficSignSymbolValue [0..1]
+ signText: LanguageCodedText [0..1]
+ signValue: Measure [0..1]
+ directionCategory: DirectionCategoryValue [0..1]
+ exitNumber: LanguageCodedText [0..1]
+ destinationInfo: DestinationInfoOnTrafficSign [0..1]
+ otherTextContent: LanguageCodedText [0..1]
«FeatureType»
Public Transport::StopPoint
+ accessibility: FeatureAccessibility [0..1]
+ publicTransportMode: PublicTransportModeValue [0..*]
+ undergroundFlag: UndergroundFlagValue [0..1]
«FeatureType»
Public Transport::RoutePoint
«FeatureType»
Road Furniture::PedestrianCrossing
+ displayClass: DisplayClassValue [0..1]
+ accessibility: FeatureAccessibility [0..1]
+ priority: PedestrianCrossingPriorityValue [0..1]
+ signage: PedestrianCrossingSignageValue [0..1]
+ type: PedestrianCrossingTypeValue [0..1]
+ level: PedestrianCrossingLevelValue [0..1]
«FeatureType»
Road Furniture::Lighting

**Figure 5.** Examples of classes based on ISO 14825 Geographic Data Files (GDF) with reused attributes.

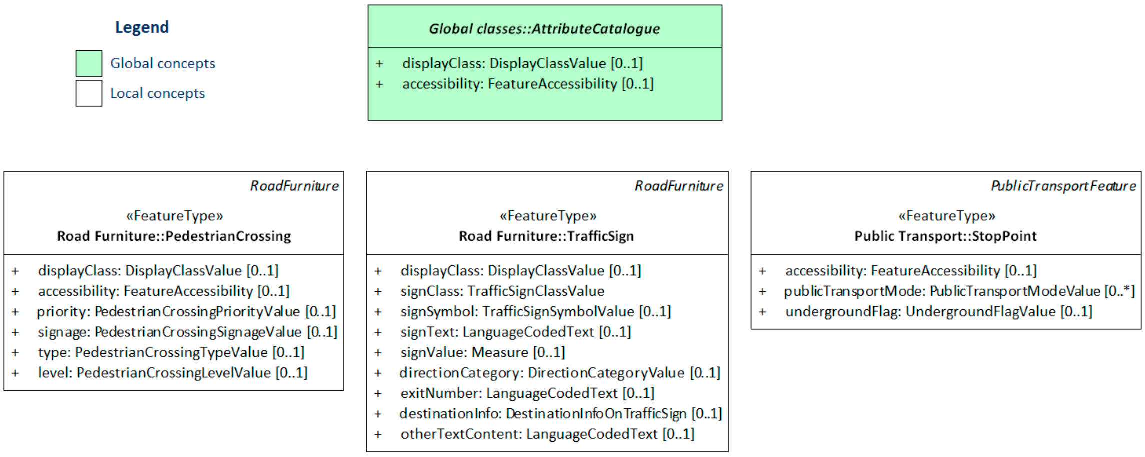

Legend
🟩 Global concepts
□ Local concepts
Global classes::AttributeCatalogue
+ displayClass: DisplayClassValue [0..1]
+ accessibility: FeatureAccessibility [0..1]
RoadFurniture
«FeatureType»
Road Furniture::PedestrianCrossing
+ displayClass: DisplayClassValue [0..1]
+ accessibility: FeatureAccessibility [0..1]
+ priority: PedestrianCrossingPriorityValue [0..1]
+ signage: PedestrianCrossingSignageValue [0..1]
+ type: PedestrianCrossingTypeValue [0..1]
+ level: PedestrianCrossingLevelValue [0..1]
RoadFurniture
«FeatureType»
Road Furniture::TrafficSign
+ displayClass: DisplayClassValue [0..1]
+ signClass: TrafficSignClassValue
+ signSymbol: TrafficSignSymbolValue [0..1]
+ signText: LanguageCodedText [0..1]
+ signValue: Measure [0..1]
+ directionCategory: DirectionCategoryValue [0..1]
+ exitNumber: LanguageCodedText [0..1]
+ destinationInfo: DestinationInfoOnTrafficSign [0..1]
+ otherTextContent: LanguageCodedText [0..1]
PublicTransportFeature
«FeatureType»
Public Transport::StopPoint
+ accessibility: FeatureAccessibility [0..1]
+ publicTransportMode: PublicTransportModeValue [0..*]
+ undergroundFlag: UndergroundFlagValue [0..1]

**Figure 6.** "AttributeCatalogue" and classes with reused attributes.

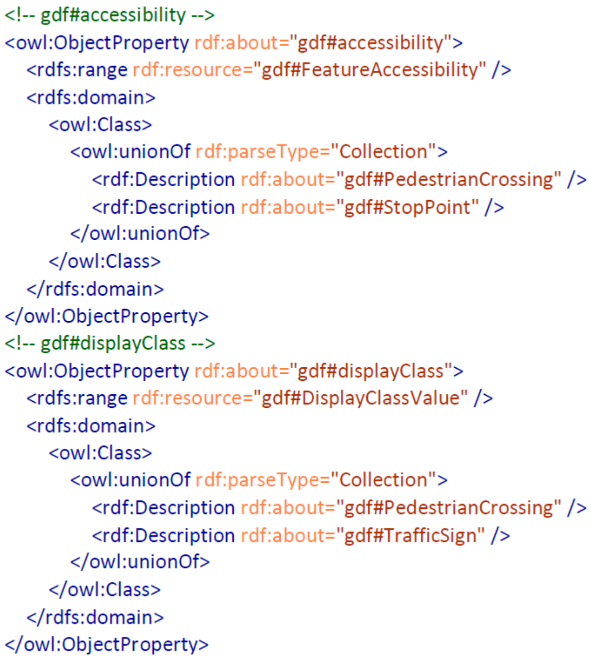
Each global attribute must be duplicated on the classes where it shall be reused as UML has no way of connecting attributes from one class to another class. Global attributes are assigned with two tagged values in the "AttributeCatalogue" class as well as in the classes where they are reused, as illustrated in Table 5. The tagged value "isGlobal" identifies whether the attributes are global or not, while "URI" stores the global identifier that connects the original and the reused copies. A script can keep all reused copies of the attributes updated from the originals in "AttributeCatalogue", by referring to the "URI" tagged value.

The conversion process from UML to OWL can use the tagged values to identify the reuse of global attributes. The attributes are converted to OWL properties and assigned with a domain, where the domain is a union of the UML classes that reuse the attributes. The conversion in this study was done manually, but a future improvement would be to implement the conversion in ShapeChange, as suggested in OGC Testbed 12.

Figure 7 shows an extract of the OWL ontology with the domain assignment for the two attributes "displayClass" and "accessibility". Figure 8 shows a graphical view of the three classes with the two superclasses, the two unions and properties.

**Table 5.** Tagged values for attributes from Figure 6.

| Attribute Name | Tagged Values | |
|---|---|---|
| | **isGlobal** | **URI** |
| AttributeCatalogue.displayClass | true | gdf#displayClass |
| AttributeCatalogue.accessibility | true | gdf#accessibility |
| PedestrianCrossing.displayClass | true | gdf#displayClass |
| PedestrianCrossing.accessibility | true | gdf#accessibility |
| PedestrianCrossing.priority | false | |
| PedestrianCrossing.signage | false | |
| PedestrianCrossing.type | false | |
| PedestrianCrossing.level | false | |
| TrafficSign.displayClass | true | gdf#displayClass |
| TrafficSign.signClass | false | |
| TrafficSign.signSymbol | false | |
| TrafficSign.signText | false | |
| TrafficSign.signValue | false | |
| TrafficSign.directionCategory | false | |
| TrafficSign.exitNumber | false | |
| TrafficSign.destinationInfo | false | |
| TrafficSign.otherTextContent | false | |
| StopPoint.accessibility | true | gdf#accessibility |
| StopPoint.publicTransportMode | false | |
| StopPoint.undergroundFlag | false | |



```
<!-- gdf#accessibility -->
<owl:ObjectProperty rdf:about="gdf#accessibility">
   <rdfs:range rdf:resource="gdf#FeatureAccessibility" />
   <rdfs:domain>
      <owl:Class>
         <owl:unionOf rdf:parseType="Collection">
            <rdf:Description rdf:about="gdf#PedestrianCrossing" />
            <rdf:Description rdf:about="gdf#StopPoint" />
         </owl:unionOf>
      </owl:Class>
   </rdfs:domain>
</owl:ObjectProperty>
<!-- gdf#displayClass -->
<owl:ObjectProperty rdf:about="gdf#displayClass">
   <rdfs:range rdf:resource="gdf#DisplayClassValue" />
   <rdfs:domain>
      <owl:Class>
         <owl:unionOf rdf:parseType="Collection">
            <rdf:Description rdf:about="gdf#PedestrianCrossing" />
            <rdf:Description rdf:about="gdf#TrafficSign" />
         </owl:unionOf>
      </owl:Class>
   </rdfs:domain>
</owl:ObjectProperty>
```

**Figure 7.** OWL domain assignment for reused properties.

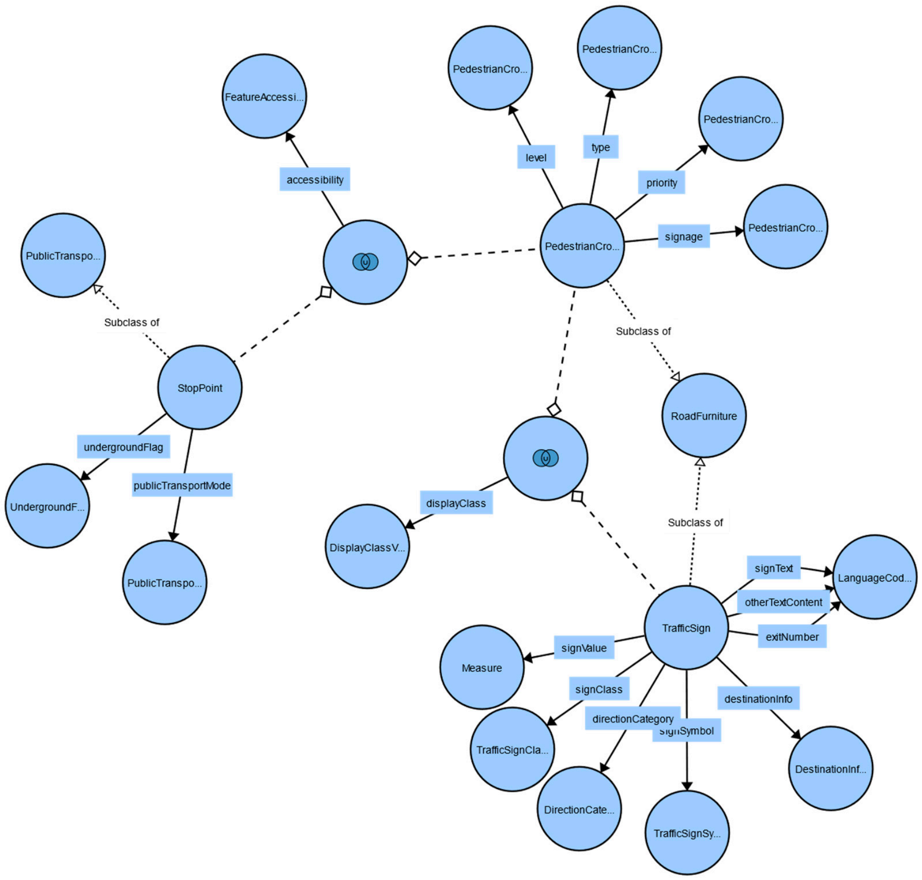


**Figure 8.** Graphical view of the OWL implementation of Figure 6 (generated with WebVOWL [79]).

### 3.4. Linking to External Concepts

Mapping of data types and classes to existing external concepts was discussed in OGC Testbed 12 and has been implemented in ShapeChange with configurable settings where the classes that shall be mapped are specified in configuration files. A method for linking UML concepts to external concepts was suggested in [44], where elements were linked to existing ontologies with tagged values. They implemented the conversion in ShapeChange through an extension. Our solution follows up the method from [44] and includes mapping to external concepts as well as linking. The tagged value "vocabulary" specifies the URI of an external concept that the internal concept shall be mapped to, while "rdfStatement" can contain an RDF statement that links the internal concept to an external concept. The tagged value "vocabulary" was specified in the INSPIRE Generic Conceptual Model [77] and is already implemented in ShapeChange rules.

Figure 9 shows the class "TrafficSign" from Figure 6 with the data type "LanguageCodedText", which is the data type for three attributes in the class: "signText", "exitNumber" and "otherTextContent". The data type "LanguageCodedText" is designed for the language-specific representation of a text, which is a common issue in information models from many domains. One example of a data type that resembles "LanguageCodedText" is the class "GeographicName" from the INSPIRE Specification for Geographical Names [80]. The OWL representation of "GeographicName" was discussed in the AR3NA project [81] where the INSPIRE Guidelines for the RDF encoding of spatial data [37] was developed. The Guidelines states that the class "GeographicName" can be simplified to the concept "rdfs:Literal" if only the string is needed, and to the subclass "rdfs:langString" if the language is needed. A related approach was used in [44], where the RDF statement "owl:equivalentProperty = rdfs:label" was added to a "name" attribute as a tagged value. The range of the "rdfs:label" property is "rdfs:Literal", which links the "name" attribute to the "rdfs:Literal" and "rdfs:langString" concepts.

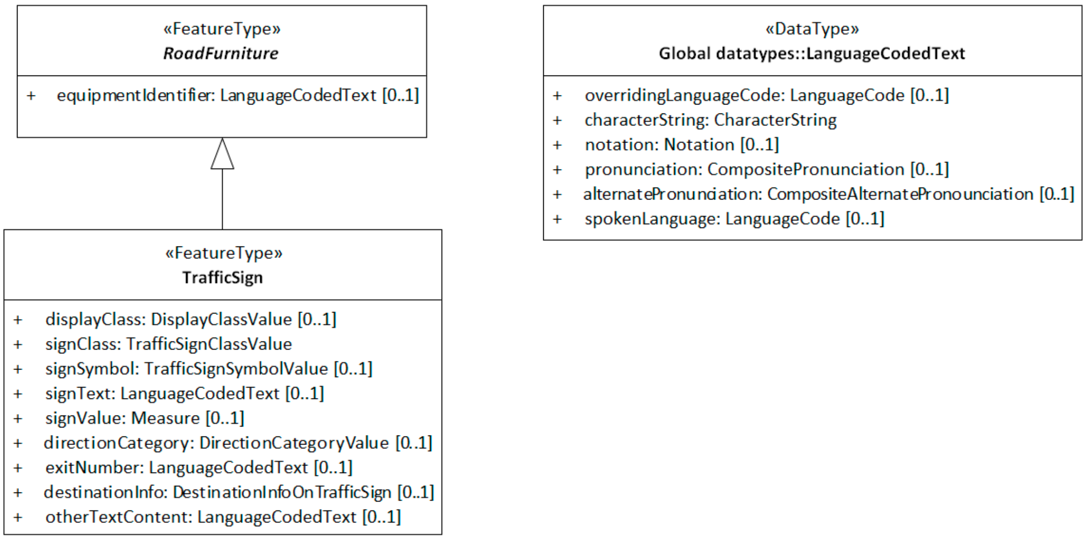

«FeatureType»
RoadFurniture
+ equipmentIdentifier: LanguageCodedText [0..1]
«FeatureType»
TrafficSign
+ displayClass: DisplayClassValue [0..1]
+ signClass: TrafficSignClassValue
+ signSymbol: TrafficSignSymbolValue [0..1]
+ signText: LanguageCodedText [0..1]
+ signValue: Measure [0..1]
+ directionCategory: DirectionCategoryValue [0..1]
+ exitNumber: LanguageCodedText [0..1]
+ destinationInfo: DestinationInfoOnTrafficSign [0..1]
+ otherTextContent: LanguageCodedText [0..1]
«DataType»
Global datatypes::LanguageCodedText
+ overridingLanguageCode: LanguageCode [0..1]
+ characterString: CharacterString
+ notation: Notation [0..1]
+ pronunciation: CompositePronunciation [0..1]
+ alternatePronunciation: CompositeAlternatePronounciation [0..1]
+ spokenLanguage: LanguageCode [0..1]

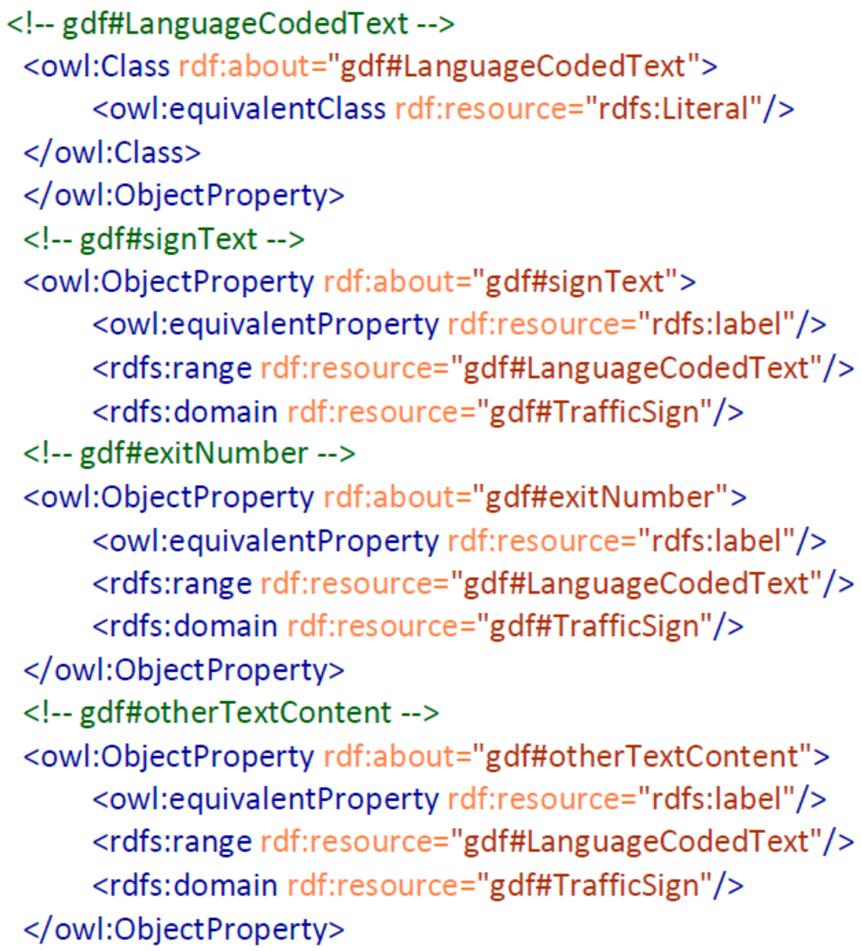
**Figure 9.** The class "TrafficSign" and the data type "LanguageCodedText".

Links from the language coded attributes and the data type "LanguageCodedText" in Figure 9 to general RDFS concepts can be specified at two levels: Each attribute can be linked to the property "rdfs:label" as in [44], while the data type can be linked to the class "rdfs:Literal". A link from the data type will be more generic and link all attributes with this data type to the external concept. A link from each attribute will reduce the number of levels needed to link the specific model to the external concept but will also require more work on specifying the RDF statements. Furthermore, links at the attribute level may also be used to separate attributes that can be linked from those that need more specific characteristics from the internal data type. Examples of RDF statements for linking attributes and data types to external concepts are shown in Table 6, while Figure 10 shows the implementation in OWL.

The "equivalent" statements in Table 6 and Figure 10 imply that classes or properties represent the same concepts but are not necessarily equal. If the data type had been equal to an external concept, the link could be specified tighter with the equality statement "sameAs". Other RDF statements such as "subclassOf" or "subPropertyOf" may also be specified, depending on the relation between the internal and external concepts.

**Table 6.** The tagged value "rdfStatement" for classes and attributes from Figure 9.

| Class or Attribute Name | Tagged Value "rdfStatement" |
|---|---|
| LanguageCodedText | owl:equivalentClass rdf:resource = "rdfs:Literal" |
| TrafficSign | |
| TrafficSign.displayClass | |
| TrafficSign.signClass | |
| TrafficSign.signSymbol | |
| TrafficSign.signText | owl:equivalentProperty rdf:resource = "rdfs:label" |
| TrafficSign.signValue | |
| TrafficSign.directionCategory | |
| TrafficSign.exitNumber | owl:equivalentProperty rdf:resource = "rdfs:label" |
| TrafficSign.destinationInfo | |
| TrafficSign.otherTextContent | owl:equivalentProperty rdf:resource = "rdfs:label" |



```
<!-- gdf#LanguageCodedText -->
 <owl:Class rdf:about="gdf#LanguageCodedText">
      <owl:equivalentClass rdf:resource="rdfs:Literal"/>
 </owl:Class>
 </owl:ObjectProperty>
 <!-- gdf#signText -->
 <owl:ObjectProperty rdf:about="gdf#signText">
      <owl:equivalentProperty rdf:resource="rdfs:label"/>
      <rdfs:range rdf:resource="gdf#LanguageCodedText"/>
      <rdfs:domain rdf:resource="gdf#TrafficSign"/>
 <!-- gdf#exitNumber -->
 <owl:ObjectProperty rdf:about="gdf#exitNumber">
      <owl:equivalentProperty rdf:resource="rdfs:label"/>
      <rdfs:range rdf:resource="gdf#LanguageCodedText"/>
      <rdfs:domain rdf:resource="gdf#TrafficSign"/>
 </owl:ObjectProperty>
 <!-- gdf#otherTextContent -->
 <owl:ObjectProperty rdf:about="gdf#otherTextContent">
      <owl:equivalentProperty rdf:resource="rdfs:label"/>
      <rdfs:range rdf:resource="gdf#LanguageCodedText"/>
      <rdfs:domain rdf:resource="gdf#TrafficSign"/>
 </owl:ObjectProperty>
```

**Figure 10.** OWL Implementation of the tagged value “rdfStatement”.

An alternative to defining equality with the “sameAs” statement is to specify reuse of external concepts in the UML model. The internal concept will then be a copy of the external concept, included in the UML model only to make the model complete for implementation in other formats than OWL. The code lists “LanguageCode” for the attribute “overridingLanguageCode” in the data type “LanguageCodedText” is an example of a data type that might be reused from external concepts. The code list is defined in an annex of the GDF standard and contains three-letter language codes according to the ISO standard ISO 639-2 [82]. The Library of Congress is the registration authority for ISO 639-2, and have made available a vocabulary for the language codes [83]. A link to the external vocabulary can be added to the UML model with the tagged value “vocabulary”. Table 7 shows the tagged values for the code list.

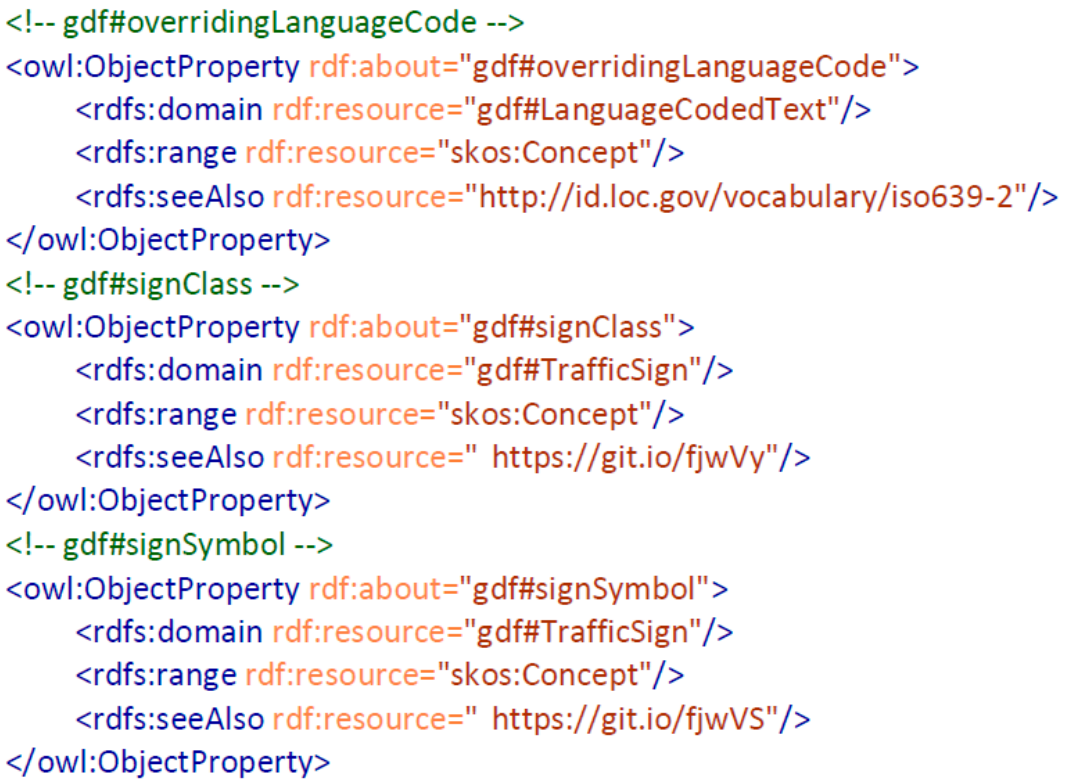
**Table 7.** Tagged value “vocabulary”.

| Class Name | Tagged Value “Vocabulary” |
|---|---|
| LanguageCode | http://id.loc.gov/vocabulary/iso639-2 |
| TrafficSignClassValue | https://git.io/fjwVy |
| TrafficSignSymbolValue | https://git.io/fjwVS |

Two attributes in the class “TrafficSign” have data types that are candidates for the reuse of external concepts as well. These are the attributes “signClass” with code list “TrafficSignClassValue” and “signSymbol” with code list “TrafficSignSymbolValue”. The standard ISO 14823 [84] defines a data dictionary for traffic signs with sign classes (service categories) and symbols (pictogram category).

These classifications should be reused in other models with traffic signs. There is no official vocabulary derived from the standard, but SKOS Concept Schemes has been developed for this study [85]. Links can be added to the UML model in the tagged value "vocabulary". Table 7 shows the tagged values for the two code lists.

Figure 11 shows the implementation in OWL for the three code lists. The OWL implementation has been done manually for the example in this study. However, conversion from UML code lists to vocabulary references in OWL are implemented in ShapeChange and have been used for the INSPIRE ontologies. The ranges of the properties are set to the generic concept "skos:Concept", while the statement "seeAlso" provides a link to the vocabulary, like in the INSPIRE ontologies.



```
<!-- gdf#overridingLanguageCode -->
<owl:ObjectProperty rdf:about="gdf#overridingLanguageCode">
      <rdfs:domain rdf:resource="gdf#LanguageCodedText"/>
      <rdfs:range rdf:resource="skos:Concept"/>
      <rdfs:seeAlso rdf:resource="http://id.loc.gov/vocabulary/iso639-2"/>
</owl:ObjectProperty>
<!-- gdf#signClass -->
<owl:ObjectProperty rdf:about="gdf#signClass">
      <rdfs:domain rdf:resource="gdf#TrafficSign"/>
      <rdfs:range rdf:resource="skos:Concept"/>
      <rdfs:seeAlso rdf:resource=" https://git.io/fjwVy"/>
</owl:ObjectProperty>
<!-- gdf#signSymbol -->
<owl:ObjectProperty rdf:about="gdf#signSymbol">
      <rdfs:domain rdf:resource="gdf#TrafficSign"/>
      <rdfs:range rdf:resource="skos:Concept"/>
      <rdfs:seeAlso rdf:resource=" https://git.io/fjwVS"/>
</owl:ObjectProperty>
```

**Figure 11.** OWL Implementation of the tagged value "vocabulary".

## 4. Discussion

### *4.1. The Scope of Ontologies for Geospatial Information*

The first and fundamental step in ontology development is to determine the domain and scope of the ontology. Defining the scope includes identifying what the ontology shall be used for and what type of questions the ontology shall answer [86]. For conversion from UML to OWL, the scope of the ontology is essential for deciding whether all restrictions from the closed UML world need to be maintained in the open OWL world, and for deciding whether or not similar concepts from existing ontologies can be reused.

We identified three main scopes for geospatial ontologies in [1] and will refer to them in further discussions as levels of information flow for which the ontologies may be used:

1. Use in Semantic Web technology and applications only.
2. Unidirectional information exchange from GIS applications to the Semantic Web.
3. Bidirectional information exchange between GIS applications and the Semantic Web.

The first level implies a decoupling from existing GIS models and applications, in which case restrictions are little needed, and the ontologies can comply with the OWA. This is not a likely situation for most of the structured geospatial information on a short term basis, as GIS databases and applications have complex and advanced functionality that will not be easily replaced with Semantic Web applications [23,34].

The second level covers the most discussed scopes (e.g., in [31,34,37]), where the original information is maintained in GIS databases and transformed to RDF for publication on the Semantic Web. Some restrictions must be applied to describe the information distinctly, but the most complex restrictions are not needed, as the ontologies will only describe existing information.

The third level implies the most complete conversion and a strict need for maintenance of restrictions. Most restrictions from UML should be maintained in the OWL implementation to ensure that the information is valid according to the closed schema in GIS databases. An example of bidirectional exchange was described in [5,6], where information was exchanged between application domains, stakeholders and lifecycle phases for construction and asset management.

Model-driven conversions from UML to OWL need to consider the different levels of information flow and must be configured according to the scope of the ontology.

### *4.2. Challenges for Conversions from UML to OWL*

The first research question in this study asked for the primary challenges in conversions from UML models of geospatial information to OWL ontologies. The results from the state-of-the-art study indicate that conversion rules handle the basic concepts packages, classes, generalizations, primitive data types, spatial data types, enumerations and simple associations consistent and thorough. Conversion of abstract classes, unions, compositions and code lists are handled with several methods that maintain the UML restrictions to various degrees. The choice of conversion method for these concepts will depend on the scope of the resulting ontologies and the level of information flow. Maintaining all restrictions require more complicated conversion methods and is not necessary for all levels. Table 8 describes conversion challenges and discusses possible solutions for the different levels of information flow.

### *4.3. Rules for UML Modelling*

The second research question in the study asked for adapted rules for UML modelling that could improve the conversion from UML models to OWL ontologies. The state-of-the-art study indicates that no standards have defined specific rules for semantics needed in UML for conversion to OWL. Two specific challenges have been pointed out in this study as candidates for improvements: conversion from UML attributes to global properties in OWL and linking to existing external concepts. We suggest an extended UML profile for geospatial information, which includes semantics for improved conversion from UML to OWL. Furthermore, Table 9 discusses rules and recommendations to ensure better implementation of UML models of geospatial information in OWL.

**Table 8.** Conversion challenges.

| Concept | Level of Information Flow | Description | Discussion |
|---|---|---|---|
| Abstract classes | 1 and 2 | No restrictions needed. | The information given by the "isAbstract" annotation in ISO 19150-2 will be satisfactory. |
| | 3 | Restrictions must prevent instances of the classes. | None of the described conversion rules maintains the restrictions.<br>Only the properties from the abstract class should be implemented in OWL, and not the class itself. The domain of the properties should then be set to a union of the subclasses, similar to our suggested solution for reused properties. |
| Unions | 1 and 2 | No restrictions needed. | The OWL "union" defined in ISO 19150-2 will be satisfactory. |
| | 3 | Restrictions are needed to define valid data types. | Conversion of UML unions may be the most complex challenge for implementation in OWL, and several methods have been suggested. The method described in the INSPIRE Guidelines and OGC Testbed 12 maintains the restrictions of a union, but it has a complicated representation in OWL. It was questioned in OGC Testbed 12 whether Semantic Web software could handle the complexity of the solution. Two alternative and simplified solutions were suggested in OGC Testbed 12 as well. Existing literature is not clear regarding what situations the different solutions should be used for. Further studies may be needed to achieve experiences from implementations and give recommendations for when to use different solutions. |
| Compositions | 1 and 2 | No restrictions needed. | The informative "aggregationType" annotation defined in ISO 19150-2 will be satisfactory. |
| | 3 | Restrictions are needed to ensure that a part instance is related to only one whole instance. | A stricter conversion with an "InverseFunctional" restriction was suggested in [67], while [66] defined a hierarchy of association types. Combining the approaches from [66] and [67] with the annotation from ISO 19150-2 may be a better solution for maintaining the implicit restrictions. |
| Code lists | 1 and 2 | No restrictions needed. | The open approach defined in OGC Testbed 12 and implemented in INSPIRE, with an informative reference to a vocabulary statement, will be satisfactory. |
| | 3 | Restrictions should refer to valid predefined values in the model or an external vocabulary, but also be open for using other values. | The ISO/TC 211 ontologies exclude additional values, while the INSPIRE ontologies have a very open approach. The approach in INSPIRE is very flexible as the resource can be anything, but the flexibility comes with the cost of reduced possibilities for the direct use of the predefined values. A possible improvement is to combine the union approach from [72] with a closer binding to clearly defined external SKOS Concept Schemes. |

**Table 9.** Rules and recommendations for UML modelling.

| Concept | Level of Information Flow | Description | Discussion |
|---|---|---|---|
| Global properties | All | Attributes that are identical in several UML classes should be converted to global properties in OWL. | Original definitions of such attributes should be maintained in one specific and abstract UML class. The class is called "AttributeCatalogue" in this study. The global attributes are reused in individual classes as copies of the original from "AttributeCatalogue". The class shall not be implemented in OWL, but the attributes in the class are implemented as global properties.<br>A specific class for attributes that can be reused in other classes is a valuable approach independently of implementation in OWL as well. Models become easier to understand and implement when identical characteristics of different real-world features are defined in a harmonized manner. |
| | All | Identification of globally defined attributes. | The tagged value "isGlobal" shall identify the attribute as global.<br>The tagged value "URI" shall be used to uniquely identify each attribute and link originals and reused copies. |
| | 3 | Restrictions must ensure that properties are assigned to specific classes. | Properties shall be linked to specific classes through a domain which is restricted to a union of the involved classes. |
| | All | Names are converted to URIs in OWL.<br>UML property names are unique within each class, while OWL properties are globally scoped. | Properties (attributes and association roles) should have unique names within a UML package that shall be implemented as an ontology.<br>This recommendation is stated in ISO 19109 with identifier /rec/general/property-name as well. |
| | All | Names are converted to URIs in OWL, and URIs are not always treated as case-sensitive. | Names of properties and classes should be non-case-sensitive unique. An example from the UML model used in the study is that the code lists have been given a suffix "Value" (e.g., "DisplayClassValue" for the attribute "displayClass"). |
| Reuse of external concepts | All | Reuse of existing concepts is a vital part of information modelling for the Semantic Web [86]. | Information modelling in UML is conducted in a closed environment and depends on concepts available in the model. However, reuse of existing concepts is a good practice that should be applied to UML models as well. Existing concepts can be duplicated in the UML models, and links to existing externals vocabularies can be added as tagged values. |
| | All | Links to external concepts. | The tagged value "rdfStatement" shall be used for linking internal and external concepts through a valid RDF statement. |
| | All | Mapping to external concepts. | The tagged value "vocabulary" shall be used for identifying the URI of external concepts that the internal concept shall be mapped to. |
| | 3 | Precise concepts are needed. | Mapping to external concepts should be done with care at this level. The mapping might lead to a loss of information in an exchange, due to differences between UML data types and external data types. |

## 5. Conclusions and Further Work

This study has analyzed methods and rules for conversion from UML models of geospatial information to implementation as OWL Ontologies. Information models for geospatial information are developed using UML, based on standards from ISO/TC 211 and OGC. A fundament for enabling geospatial information for the Semantic Web will be to convert the UML models to OWL ontologies. A state-of-the-art study of conversion rules as described in standards and research indicates that basic concepts from UML can be converted to OWL. However, UML models of geospatial information are developed in a closed world in a geospatial context, while a core principle for OWL is the Open World Assumption (OWA). The UML concepts of abstract classes, unions, compositions and code lists represent closed world restrictions that must be treated specially to be maintained in the ontology. None of the existing conversion methods fully maintain these restrictions. Possible improvements have been suggested in this study and may be followed up in further studies.

Furthermore, attributes and associations in UML are uniquely owned by each class, while properties in OWL are globally scoped and may be used by many classes. This study suggests that global attributes should be treated specially in UML models, and specifies the use of tagged values to add semantics for converting them to global properties in OWL. Finally, an essential principle in ontology development is an extensive reuse of concepts from existing ontologies, while UML models mainly use concepts that are internally defined in the model. For this matter, mapping to existing ontologies has been defined in several ways. This study suggests a specific use of tagged values to add semantics to UML models for linking and mapping internal UML concepts to external OWL concepts.

The scope of the ontology is essential for the precision needed in the conversion. Ontologies that shall only be used within the Semantic Web or for publishing information from geospatial databases on the Semantic Web do not need to maintain all restrictions from UML and can apply mapping to external concepts. Ontologies that shall be used for updating geospatial databases with content from the Semantic Web need to be more precise and maintain both restrictions and specific internal concepts, to avoid instances that are invalid according to the UML model. Conversion settings that can be configured according to the scope of the ontology is a necessary fundament for a model-driven implementation as OWL ontologies.

**Author Contributions:** Conceptualization, K.J.; methodology, K.J., E.O. and L.H.; validation, K.J.; formal analysis, K.J.; investigation, K.J.; data curation, K.J.; writing—original draft preparation, K.J.; writing—review and editing, E.O. and L.H.; visualization, K.J.; supervision, E.O. and L.H.

**Funding:** This research received no external funding.

**Acknowledgments:** Rune Hjelsvold provided invaluable support to the state-of-the-art-study for this article. NTNU Open Access publishing funds cover the article processing charges.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jetlund, K. Improvements in Automated Derivation of owl ontologies from geospatial uml models. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2018**, *XLII-4*, 283–290. [CrossRef]
2. Dangol, A.; Dewaelheyns, V.; Steenberghen, T. Why Geospatial Linked Open Data for Smart Mobility? In *REAL CORP 2016–SMART ME UP! How to Become and How to Stay a Smart City, and Does This Improve Quality of Life? Proceedings of the 21st International Conference on Urban. Planning, Regional Development and Information Society, Hamburg, Germany, 22–24 June 2016*; CORP—Competence Center of Urban and Regional Planning: Vienna, Austria, 2016.
3. Prudhomme, C.; Homburg, T.; Ponciano, J.-J.; Boochs, F.; Cruz, C.; Roxin, A.-M. Interpretation and automatic integration of geospatial data into the Semantic Web. *Computing* **2019**, 1–27. [CrossRef]
4. Zhang, C.R.; Zhao, T.; Anselin, L.; Li, W.D.; Chen, K. A Map-Reduce based parallel approach for improving query performance in a geospatial semantic web for disaster response. *Earth Sci. Inform.* **2015**, *8*, 499–509. [CrossRef]
5. Luiten, B.; Böhms, M.; O'Keeffe, A.; van Nederveen, S.; Bakker, J.; Wikström, L. A hybrid linked data approach to support asset management. In *Life-Cycle of Engineering Systems: Emphasis on Sustainable Civil Infrastructure: Proceedings of the Fifth International Symposium on Life-Cycle Civil Engineering (IALCCE 2016)*; CRC Press, Taylor & Francis Group: London, UK, 2016.
6. Luiten, B.; O'Keeffe, A.; Stolk, S.; Wikström, L.; Weise, M. *Interlink D4. Principles for a European Road OTL*; CEDR: Brussels, Belgium, 2017.
7. Karan, E.P.; Irizarry, J. Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Autom. Constr.* **2015**, *53*, 1–12. [CrossRef]
8. Hitzler, P.; Janowicz, K.; Krisnadhi, A.A. Ontology Modeling with Domain Experts: The GeoVocamp Experience. In Proceedings of the Diversity++@ ISWC 2015, Bethlehem, PA, USA, 12 October 2015.

9. Jetlund, K.; Onstein, E.; Huang, L. Information Exchange between GIS and Geospatial ITS Databases Based on a Generic Model. *Isprs Int. J. Geo Inf.* **2019**, *8*, 141. [CrossRef]
10. Coetzee, S.; Plews, R.; Brodeur, J.; Hjelmager, J.; Jones, A.; Jetlund, K.; Grillmayer, R.; Wasström, C. Standards Making Geographic Information Discoverable, Accessible and Usable for Modern Cartography. In *Service-Oriented Mapping: Changing Paradigm in Map Production and Geoinformation Management*; Döllner, J., Jobst, M., Schmitz, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; pp. 325–344.
11. Object Management Group. *Unified Modelling Language Specification Version 2.5.1*; Object Management Group: Needham, MA, USA, 2017.
12. ISO/TC 211. *ISO 19109:2015 Geographic Information–Rules for Application Schema*; ISO: Geneva, Switzerland, 2015.
13. ISO/TC 211. *ISO 19103:2015 Geographic Information–Conceptual Schema Language*; ISO: Geneva, Switzerland, 2015.
14. Object Management Group. *Model Driven Architecture (MDA) Guide Rev. 2.0*; Object Management Group: Needham, MA, USA, 2014.
15. ISO/TC 211. *ISO 19136:2007 Geographic Information–Geography Markup Language (GML)*; ISO: Geneva, Switzerland, 2007.
16. Jeansoulin, R. Review of Forty Years of Technological Changes in Geomatics toward the Big Data Paradigm. *ISPRS Int. J. Geo Inf.* **2016**, *5*, 155. [CrossRef]
17. ISO/TC 211. *ISO 19128:2005 Geographic Information–Web Map Server Interface*; ISO: Geneva, Switzerland, 2005.
18. ISO/TC 211. *ISO 19142:2010 Geographic Information–Web Feature Service*; ISO: Geneva, Switzerland, 2010.
19. Zhang, C.R.; Zhao, T.; Li, W.D.; Osleeb, J.P. Towards logic-based geospatial feature discovery and integration using web feature service and geospatial semantic web. *Int. J. Geogr. Inf. Sci.* **2010**, *24*, 903–923. [CrossRef]
20. Iwaniak, A.; Kaczmarek, I.; Strzelecki, M.; Lukowicz, J.; Jankowski, P. Enriching and improving the quality of linked data with GIS. *Open Geosci.* **2016**, *8*, 323–336. [CrossRef]
21. Manola, F.; Miller, E.; McBride, B. RDF 1.1 Primer. W3C Working Group Note 2014. Available online: https://www.w3.org/TR/rdf11-primer/ (accessed on 21 August 2019).
22. Hitzler, P.; Krötzsch, M.; Parsia, B.; Patel-Schneider, P.F.; Rudolph, S. OWL 2 Web Ontology Language Primer (Second Edition). *W3C Recomm.* **2012**, *27*, 123.
23. Hart, G.; Dolbear, C. *Linked Data: A Geographic Perspective*; CRC Press: Boca Raton, FL, USA, 2016.
24. Aditya, T.; Kraak, M.-J. A Search Interface for an SDI: Implementation and Evaluation of Metadata Visualization Strategies. *Trans. GIS* **2007**, *11*, 413–435. [CrossRef]
25. Klien, E. A Rule-Based Strategy for the Semantic Annotation of Geodata. *Trans. GIS* **2007**, *11*, 437–452. [CrossRef]
26. Lutz, M.; Kolas, D. Rule-Based Discovery in Spatial Data Infrastructure. *Trans. GIS* **2007**, *11*, 317–336. [CrossRef]
27. Wiemann, S.; Bernard, L. Spatial data fusion in Spatial Data Infrastructures using Linked Data. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 613–636. [CrossRef]
28. Hietanen, E.; Lehto, L.; Latvala, P. Providing Geographic Datasets as Linked Data in Sdi. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 583–586. [CrossRef]
29. Patroumpas, K.; Georgomanolis, N.; Stratiotis, T.; Alexakis, M.; Athanasiou, S. Exposing INSPIRE on the Semantic Web. *J. Web Semant.* **2015**, *35*, 53–62. [CrossRef]
30. Ulutaş Karakol, D.; Kara, G.; Yılmaz, C.; Cömert, Ç. Semantic Linking Spatial Rdf Data to the Web Data Sources. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII–4*, 639–645. [CrossRef]
31. Aydinoğlu, A.Ç.; Kara, A. Modelling and publishing geographic data with model-driven and linked data approaches: Case study of administrative units in Turkey. *J. Spat. Sci.* **2019**, *64*, 11–31. [CrossRef]
32. Wang, Y.D.; Qiao, M.L.; Liu, H.; Ye, X.Y. Qualitative spatial reasoning on topological relations by combining the semantic web and constraint satisfaction. *Geo Spat. Inf. Sci.* **2018**, *21*, 80–92. [CrossRef]
33. Mobasheri, A. A Rule-Based Spatial Reasoning Approach for OpenStreetMap Data Quality Enrichment; Case Study of Routing and Navigation. *Sensors* **2017**, *17*, 2498. [CrossRef]
34. Tand, J.; van den Brink, L.; Barnaghi, P. Spatial Data on the Web Best Practices. W3C Working Group Note 2017. Available online: https://www.w3.org/TR/2017/NOTE-sdw-bp-20170928/ (accessed on 21 August 2019).
35. ISO/TC 211. *ISO 19150-2:2015 Geographic Information–Ontology–Part 2: Rules for Developing Ontologies in the Web Ontology Language (OWL)*; ISO: Geneva, Switzerland, 2015.

36. ISO/TC 211. ISO/TC 211 Group for Ontology Management. 2018. Available online: https://github.com/ISO-TC211/GOM (accessed on 15 June 2019).
37. ARE3NA Project. Guidelines for the RDF Encoding of Spatial Data. 2017. Available online: http://inspire-eu-rdf.github.io/inspire-rdf-guidelines/ (accessed on 15 June 2019).
38. Echterhoff, J.; Portele, C.; Birkel, P.; Nichols, D.L.; Badgley, E.D. *OGC Testbed-12: ShapeChange Engineering Report*; Open Geospatial Consoritum: Wayland, MA, USA, 2017.
39. Echterhoff, J.; Birkel, P.; Nichols, D.L. *OGC Testbed-14: Application Schema-Based Ontology Development Engineering Report*; Open Geospatial Consoritum: Wayland, MA, USA, 2018.
40. McGlinn, K.; Wagner, A.; Pauwels, P.; Bonsma, P.; Kelly, P.; O'Sullivan, D. Interlinking geospatial and building geometry with existing and developing standards on the web. *Autom. Constr.* **2019**, *103*, 235–250. [CrossRef]
41. Debruyne, C.; Meehan, A.; Clinton, É.; McNerney, L.; Nautiyal, A.; Lavin, P.; O'Sullivan, D. Ireland's Authoritative Geospatial Linked Data. In *International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2017.
42. Debruyne, C.; Clinton, É.; McNerney, L.; Nautiyal, A.; O'Sullivan, D. Serving Ireland's Geospatial Information as Linked Data. In Proceedings of the International Semantic Web Conference (Posters & Demos), Kobe, Japan, 17–21 October 2016.
43. Ordnance Survey. Ordnance Survey Ontologies. Available online: http://data.ordnancesurvey.co.uk/ontology (accessed on 6 March 2019).
44. Van den Brink, L.; Janssen, P.; Quak, W.; Stoter, J. Linking spatial data: Automated conversion of geo-information models and GML data to RDF. *Int. J. Spat. Data Infrastruct. Res.* **2014**, *9*, 59–85. [CrossRef]
45. Folmer, E.; Beek, W.; Rietveld, L. Linked Data Viewing as Part of the Spatial Data Platform of the Future. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII-4/W8*, 49–52. [CrossRef]
46. Vilches-Blázquez, L.M.; Villazón-Terrazas, B.; Corcho, O.; Gómez-Pérez, A. Integrating geographical information in the Linked Digital Earth. *Int. J. Digit. Earth* **2014**, *7*, 554–575. [CrossRef]
47. Greek Linked Open Data. Available online: http://linkedopendata.gr/ (accessed on 6 March 2019).
48. Allemang, D.; Hendler, J.A. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*, 2nd ed.; Elsevier: Burlington, MA, USA, 2011.
49. Bārzdiņš, J.; Bārzdiņš, G.; Čerāns, K.; Liepiņš, R.; Sproģis, A. UML style graphical notation and editor for OWL 2. In *International Conference on Business Informatics Research*; Springer: Berlin/Heidelberg, Germany, 2010.
50. Brockmans, S.; Haase, P.; Hitzler, P.; Studer, R. A metamodel and UML profile for rule-extended OWL DL ontologies. In *Semantic Web: Research and Applications, Proceedings*; Sure, Y., Domingue, J., Eds.; Springer: Berlin, Germany, 2006; pp. 303–316.
51. Brockmans, S.; Volz, R.; Eberhart, A.; Loffler, P. Visual modeling of OWL DL ontologies using UML. In *Semantic Web Iswc 2004 Proceedings*; McIlraith, S.A., Plexousakis, D., VanHarmelen, F., Eds.; Springer: Berlin, Germany, 2004; pp. 198–213.
52. Cranefield, S.; Purvis, M. A UML profile and mapping for the generation of ontology-specific content languages. *Knowl. Eng. Rev.* **2002**, *17*, 21–39. [CrossRef]
53. Djuric, D.; Gasevic, D.; Devedzic, V.; Damjanovic, V. UML profile for OWL. In *Web Engineering, Proceedings*; Koch, N., Fraternali, P., Wirsing, M., Eds.; Springer: Berlin, Germany, 2004; pp. 607–608.
54. Djuric, D.; Gasevic, D.; Devedzic, V.; Damjanovic, V. A UML profile for OWL ontologies. In *Model Driven Architecture*; Assmann, U., Aksit, M., Rensink, A., Eds.; Springer: Berlin, Germany, 2005; pp. 204–219.
55. Evermann, J.; Wand, Y. Ontological modeling rules for UML: An empirical assessment. *J. Comput. Inf. Syst.* **2006**, *46*, 14–29. [CrossRef]
56. Guizzardi, G.; Wagner, G.; Guarino, N.; van Sinderen, M. An ontologically well-founded profile for UML conceptual models. In *Advanced Information Systems Engineering, Proceedings*; Persson, A., Stirna, J., Eds.; Springer: Berlin, Germany, 2004; pp. 112–126.
57. Guizzardi, G.; Wagner, G.; Herre, H. On the foundations of UML as an ontology representation language. In *Engineering Knowledge in the Age of the Semantic Web, Proceedings*; Motta, E., Shadbolt, N., Stutt, A., Gibbins, N., Eds.; Springer: Berlin, Germany, 2004; pp. 47–62.
58. Keet, C.M.; Fillottrani, P.R. An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2. *Data Knowl. Eng.* **2015**, *98*, 30–53. [CrossRef]

59. Kogut, P.; Cranefield, S.; Hart, L.; Dutra, M.; Baclawski, K.; Kokar, M.; Smith, J. UML for ontology development. *Knowl. Eng. Rev.* **2002**, *17*, 61–64. [CrossRef]
60. Object Management Group. *Ontology Definition Metamodel, Version 1.0.*; Object Management Group: Needham, MA, USA, 2009.
61. Parreiras, F.S.; Staab, S. Using ontologies with UML class-based modeling: The TwoUse approach. *Data Knowl. Eng.* **2010**, *69*, 1194–1207. [CrossRef]
62. Interactive Instruments GmbH, ShapeChange. Available online: http://shapechange.net (accessed on 15 June 2019).
63. Bourahla, M.; Belghiat, A. UML Class Diagrams to OWL Ontologies: A Graph Transformation based Approach. *Int. J. Comput. Appl.* **2012**, *41*, 41–46. [CrossRef]
64. Bourahla, M.; Belghiat, A. Transformation of UML models towards OWL ontologies. In Proceedings of the 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, SETIT, Sousse, Tunisia, 21–24 March 2012.
65. Xu, Z.M.; Ni, Y.Y.; He, W.J.; Lin, L.L.; Yan, Q. Automatic extraction of OWL ontologies from UML class diagrams: A semantics-preserving approach. *World Wide Web Internet Web Inf. Syst.* **2012**, *15*, 517–545. [CrossRef]
66. Bahaj, M.; Bakkas, J. Automatic Conversion Method of Class Diagrams to Ontologies Maintaining Their Semantic Features. *Int. J. Soft Comput. Eng. (IJSCE)* **2013**, *2*, 65–69.
67. Hajjamy, O.E.; Alaoui, K.; Alaoui, L.; Bahaj, M. Mapping UML to OWL2 ontology. *J. Theor. Appl. Inf. Technol.* **2016**, *90*, 126–143.
68. Gasevic, D.; Djuric, D.; Devedzic, V.; Damjanovi, V. Converting UML to OWL ontologies. In Proceedings of the 13th international World Wide Web conference on Alternate Track Papers & Posters WWW Alt. '04, New York, NY, USA, 19–21 May 2004.
69. Gherabi, N.; Bahaj, M. A New Method for Mapping UML Class into OWL Ontology. *Int. J. Comput. Appl.* **2012**, *1*, 5–9.
70. Probst, F.; Bibotti, F.; Pazos, A. Connecting ISO and OGC Models to the Semantic Web | OGC Network. In Proceedings of the 3rd International Conference on Geographic Information Science: Extended Abstracts and Poster Summaries, Adelphi, MD, USA, 20–23 October 2004; pp. 181–184.
71. Buccella, A.; Cechich, A.; Gendarmi, D.; Lanubile, F.; Semeraro, G.; Colagrossi, A. Building a global normalized ontology for integrating geographic data sources. *Comput. Geosci.* **2011**, *37*, 893–916. [CrossRef]
72. Zedlitz, J.; Luttenberger, N. Transforming Between UML Conceptual Models And OWL 2 Ontologies. In Proceedings of the Terra Cognita Workshop on Foundations Technologies and Applications of the Geospatial Web at the 11th International Semantic Web Conference ISWC 2012, Boston, MA, USA, 12 November 2012.
73. ISO/TC 211. *ISO 19107:2003 Geographic Information–Spatial Schema*; ISO: Geneva, Switzerland, 2003.
74. Perry, M.; Herring, J. *OGC GeoSPARQL-A Geographic Query Language for RDF Data*; Open Geospatial Concortium: Wayland, MA, USA, 2012.
75. Cox, S. An explicit OWL representation of ISO/OGC Observations and Measurements. *SSN@ ISWC* **2013**, *1063*, 1–18.
76. ISO/TC 211. *ISO 19125-1:2004 Geographic Information–Simple Feature Access–Part 1: Common Architecture*; ISO: Geneva, Switzerland, 2004.
77. Inspire. *Inspire Generic Conceptual Model*; INSPIRE Drafting Team "Data Specification": Brussels, Belgium, 2013.
78. ISO/TC 204. *ISO 14825:2011 Intelligent Transport Systems–Geographic Data Files (GDF)–GDF5.0*; ISO: Geneva, Switzerland, 2011.
79. Link, V.; Lohmann, S.; Marbach, E.; Negru, S.; Wiens, V. WebVOWL. 2019. Available online: http://www.visualdataweb.de/webvowl. (accessed on 21 August 2019).
80. Inspire. *Inspire Data Specification on Geographical Names–Guidelines*; INSPIRE Thematic Working Group Geogrphical Names: Brussels, Belgium, 2010.
81. Echterhoff, J.; De Paepe, D. Encoding of Geographical Names. 2017. Available online: https://github.com/inspire-eu-rdf/inspire-rdf-guidelines/issues/28 (accessed on 17 June 2019).
82. ISO/TC 37/SC 2. *ISO 639-2:1998 Codes for the Representation of Names of Languages–Part 2: Alpha-3 Code*; ISO: Geneva, Switzerland, 1998.

83. The Library of Congress, ISO639-2 Languages. 2019. Available online: http://id.loc.gov/vocabulary/iso639-2 (accessed on 21 August 2019).
84. ISO/TC 204. *ISO 14823:2017 Intelligent Transport Systems–Graphic Data Dictionary*; ISO: Geneva, Switzerland, 2017.
85. Jetlund, K. ISO 14823 SKOS Concept Schemes. 2019. Available online: https://github.com/jetgeo/GIS2OWL/tree/master/iso14823 (accessed on 21 August 2019).
86. Noy, N.F.; McGuinness, D.L. Ontology Development 101: A Guide to Creating Your First Ontology. 2001. Available online: https://protegewiki.stanford.edu/wiki/Ontology101 (accessed on 21 August 2019).