# **Strategies for Publishing Domain Ontologies as Linked Data from OGC domain standards**

## **Authors**

|  |  |
| --- | --- |
| **Name** | **Affiliation** |
| Sylvain Grellet | BRGM |
| Abdelfettah Feliachi | ATOS |
| Rob Atkinson | OGC |
| Eric Boisvert | GSC |
| Marcus Sen | BGS |
| Mickaël Beaufils | BRGM |
| Alistair Ritchie | Manaaki Whenua – Landcare Research |
| James Passmore | BGS |
| Boyan Brodaric | GSC |
| Katharina Schleidt | Data Cove |
| Chuck Heazel | WiSC Enterprises |
| Steve Richard | AZGS |

## 

## **Introduction**

OGC Environmental Linked Features Interoperability Experiment (ELFIE) has explored publishing environmental observations as linked data, with data interpretation supported by having unambiguous references to key domain concepts in the data represented by additional links (URIs). Such concepts cover the nature of the properties of a data object, but also the key Use Case of linking an observation to the Feature of Interest.

Parallel to this activity, members from several OGC SWGs (GeoSciML, Groundwater, Hydrologic Features) are attempting to ‘port’ the initial work done using ISO 191xx series UML paradigm to OWL.

Taking the opportunity of having all the involved parties available during the 2018 Orléans TC an ad’hoc meeting was organized on March 2018 23rd.

This document builds on those joint discussions in order to share best practices. It is shared under GeoSciML SWG GitHub repository (<https://github.com/opengeospatial/GeoSciML/tree/master/documents> ).

## **Problem statement**

Common semantics for the description of key domain concepts and the type of observations’ the Feature of Interest are identified by using published models – such as HY\_Features, GWML, GeoSciML. These models (currently) are available in a normative forms as UML notation, and in some cases as XML schema, neither of which is conducive to easily dereference a URI for a concept to get more detail about that specific concept, or even to confidently match the identity of any references to concepts.

The requirement therefore is to be able to publish such models, as stable URIs, dereferencing as fine-grained Linked Data. This means that a predictable URI naming scheme for elements in a model is required. (this may not be strictly necessary for a single model, but when a body of such models are managed by a single authority – such as OGC and its delegated Specification Working Groups – then commonality is necessary to avoid revisiting the naming strategy for each case, and to allow users to become familiar and comfortable with a consistent product.)

Within ELFIE context the need has been identified for at least two models, HY\_Features and GWML2, and their multiple component Application Schemas. The GeoSciML community is also looking at the same issues. There is a need for both efficiency and consistency to harmonise and document a common approach.

ISO 19150-2 defines a set of rules for encoding UML as an OWL ontology referencing the suite of OWL artefacts created to model the ISO Harmonised Model.

This form of OWL has an unknown utility – by tying data types and concepts into the extensive ISO model this adds a high burden on clients to reason over the entire model (quite a large set of content) in order to identify fairly simple, but highly important, semantic baseline information – such as that a property may be treated as a xsd:integer for purposes of a calculation). The feasibility of such reasoning is unknown, as there is no way yet to test the ISO models behave as expected under reasoning conditions.

Furthermore, for the purposes of Linked Data, much of the modelling of behavior of low-level data types is not relevant, when the main requirement is to identify concepts, access explanations, or potentially access information about implementations of these concepts that may be available.

Finally, from the perspective of infrastructure support, OGC provides a “definitions server” that is designed around SKOS meta-model and will provide dereferencing services and Linked Data representations.

## **Background**

BRGM and GSC have performed an initial analysis of the issues and questions that arise in the context of:

“· Generating ontologies for a subset of GeoSciML: GeoSciML Basic, Borehole and Lite,

· Testing populating instances for Boreholes, Geologic Units pointing to vocabs when available (FR, EU, CGI GTWG) and testing some representations (maps, graph).”

The output of that exercise has been shared under BRGM repository (<https://github.com/BRGM/GeoSciMLontology/blob/master/documents/ISO191xx_2_OWL_NoteBRGM.docx>)

Identified issues and questions are relevant to the equivalent challenges for ELFIE, and collaboration with the GeoSciML, GWML, HY\_Features and ELFIE activities has been initiated to develop a common solution. Those elements have then been further characterized from the perspective of definition publication process and governance.

Building on this during the 2018 Orléans TC ad’hoc meeting on March 23rd, identified issues were discussed.

**Proposed strategy**

Resources, time and testing methodology for “hand building” optimal equivalent ontologies are not available (notwithstanding comments that these may lead to iterative refinement and improvement of the model).

The only automated pathway available to all participants currently is using Shapechange (<http://shapechange.net>), which means developing a set of encoding rules from a wide range of choices and options from previous experiments in encoding different styles of UML model.

The ISO 19150-2 artefacts will be seen as an intermediate artefact available by an annotation property reference in the metadata describing the provenance of the ontology. For such perspective the use of PROV-O should be used.

“Manual interventions” to artefact to correct bugs or apply further rules will be encoded as RDF transformations or additional statements that can be applied in sequence.

The process can be summarized like this:

* 1°/ ShapeChange starting point : conceptual/logical model
* 2°/ manual adjustments according to what is written in this document

If necessary, rules to extract basic OWL and xsd: datatype equivalence, without embedded reference to ISO datatypes will be developed and applied as part of OGC-NA infrastructure.

Rules to extract SKOS equivalent glossary/taxonomy from OWL class models will be developed and applied as part of OGC-NA infrastructure.

URIs under [www.opengis.net/def](http://www.opengis.net/def) will be assigned, and content will be marked as draft, as an exercise in publication governance by OGC-NA, to make content and process available for wider discussion.

## **Issues identified**

### **Naming Policy**

OGC published ontologies will have URIs and Linked Data access based on identifiers under a pattern yet to be chosen.

*Note:*

* Words between curly brackets (ex:{ontology}) are ‘reserved words’ thus will remain as is when applied in URIs
* Words between angle brackets will be replaced by the corresponding values when applied in URIs (ex : ‘hyf’, ‘gwml2’, ‘gsml’ for ex:<schemaAcronym>)
* as <http://www.opengis.net/def/> and <http://www.opengis.net/def/auth/> resolve to a wide variety of different notions
* writing convention
  + Class names will be UpperCamelCase names e.g. **s1:Class1**
  + Properties will be lowerCamelCase e.g. **s1:prop1.**

Except for class scoped properties which names are ambiguous (ex : 2 classes having homonym properties but with different semantics) where the applied formalism will be **s1:Class1.prop1**

* + For more details: see below
* General semantic web BP
  + base/document/ for identifying informative resources
  + base/id/ for identifying real world entities
  + base/def/ for identifying ontologies and their components

Option 1: @prefix s1: http(s)://[www.opengis.net/def/](http://www.opengis.net/def/){ontology}/<authority>/<schemaAcronym>

ex: [http(s)://www.opengis.net/def/ontology/HydroDWG/hyf](https://www.opengis.net/def/ontology/HydroDWG/hyf/) or [http(s)://www.opengis.net/def/ontology/HydrologicFeaturesSWG/hyf](https://www.opengis.net/def/ontology/HydroDWG/hyf/)

Option 2: http(s)://[www.opengis.net/def/](http://www.opengis.net/def/)<authority>/{ontology}/<schemaAcronym>

ex: [http(s)://www.opengis.net/def/HydroDWG/ontology/hyf](https://www.opengis.net/def/ontology/HydroDWG/hyf/) or

[http(s)://www.opengis.net/def/HydrologicFeaturesSWG/ontology/hyf](https://www.opengis.net/def/ontology/HydroDWG/hyf/)

Option 3: http(s)://[www.opengis.net/def](http://www.opengis.net/def/)/{ontology}/<schemaAcronym>

ex: [http(s)://www.opengis.net/def/ontology/hyf](https://www.opengis.net/def/ontology/HydroDWG/hyf/)

[http(s)://www.opengis.net/def/ontology/hyf/HY\_Waterbody](https://www.opengis.net/def//ontology/hyf/HY_Waterbody) -> will be the identifier of the class in the ontology

Option 4: http(s)://[www.opengis.net/def](http://www.opengis.net/def/)/<schemaAcronym>

ex: [http(s)://www.opengis.net/def/hyf](https://www.opengis.net/def/ontology/HydroDWG/hyf/) ([http(s)://www.opengis.net/def/hyf/HY\_Waterbody](https://www.opengis.net/def//ontology/hyf/HY_Waterbody))

[http(s)://www.opengis.net/def/gwml2](https://www.opengis.net/def/ontology/HydroDWG/hyf/)

[http(s)://www.opengis.net/def/gsml](https://www.opengis.net/def/ontology/HydroDWG/hyf/)

Under that option, the reserved word {ontology} being removed, the client has to specify which representation is desired. Thus owl model, rdf, xsd, json-ld context will be returned based on content negotiation (Accept:header) or an explicit file extension

(ex: <https://www.opengis.net/def/gwml2.xsd> , <https://www.opengis.net/def/gwml2.ttl> ).

### **Weaknesses or issues with ISO 19150-2 rules**

* The rules of ISO 19150-2 restrict the resulting ontologies to the way UML metamodel works . Respecting all of the 19150-2 means we don’t take into account the open world assumption when working with ontologies (missing a piece of information doesn’t mean that piece of information is false). For instance, placeholder properties or classes in UML are transformed to OWL properties and classes where there is no need for them.
* The transformation rules are consistent but limits the resulting ontologies to the UML paradigm. Some additional work may be done on the resulting ontologies to add semantics between classes (disjunctions, subsumption, equivalence, etc) and within or between properties (functional properties, transitive properties, symmetric properties, inverse of, etc).
* No specific indications about association classes are mentioned in the norm. It is obvious that an association class is translated as an OWL class. No rule for linking this class to the related class(es) appear.
* Union

19150-2 recommends to use owl:UnionOf, the implementation in ShapeChange seems rather to stick to 19118 approach (disjoint union) but does it in a very complex way.

Instead, owl:disjointUnionOf shall be used

This should generate a ChangeRequest to 19150-2

**Property names and definitions**

* Properties naming when translating attributes: dots in properties identifiers could be interpreted somehow that they are still scoped to classes, while in ontologies, properties are scoped to a namespace instead. Properties are independent entities that may or may not have a specific class as a domain. This is one major structural difference between UML and OWL.
  + Use general (non-scoped to class) property names when the name of the attribute or association is unique. Thus, leave the domain of the properties open (or typed as owl:Thing). The restrictions on the properties values in the class definition can be used for this purpose.
  + When there is an ambiguïty, allow scoped names for properties (class.Property) then verify whether
    - automatically created properties can be merged into one (eg. GeologicFeature.purpose and EarthMaterial.purpose).
    - or if automatically created properties can be subProperty of a higher one.
* Domains and ranges of properties
  + Domains and ranges properties should not be defined in the reference ontology to favour reuse. They could be specified it in application ontologies that reuse the properties (if needed). Instead, restriction on the values of the properties should be defined for every class.

**Alignment documents (UML -> OWL)**

* are place to put subProperty relationships (roleA and roleB are flavours of role ) - also equivalences across application schema
* Skos:notation (datatype to be determined) to preserve original property name token - for display and reference to xpath elements
* Both are not automatised yet in ShapeChange

### **SWE related issues**

There is no direct SWE ontology but several concepts from SWE can be found elsewhere

* Reference to basic SWE types must be modified if needed by specialized Classes from other ontologies or by defining new ones.
* Use GSML\_QuantityRange instead of swe:QuantityRange as recommended in GeoSciML definition.
* Rename swe:Category to skos:Concept or mdl:Lineage (depending on the case) and swe:Quantity to the relevant class in the context (ts:TimePosition, mdq:PositionalAccuracy, etc.).

### **Preparing for application ontologies**

* To enable GeoSciML Basic and Borehole properties to be reused in application ontologies like GeoSciML Lite, we activate the ShapeChange rule “rule-owl-prop-globalScopeByUniquePropertyName” that scopes unique name property to global use, and thus not specify the domain of these properties. The scoping of the properties to their classes in Basic and Borehole is done using restrictions on the values that these properties can take for their corresponding classes. This can be done thanks to the ShapeChange rule “rule-owl-prop-range-local-withUniversalQuantification”.

### **UML constraints**

In the considered standards, constraints are expressed in non-canonical forms in the UML classes definitions or in OCL.

* The requirements of the model cannot be all respected in the ontology representation (eg. “QuantityRange properties that must report a single value SHALL assign both lower and upper value as equal to that single value.”). This should be checked and translated manually as restrictions (owl:Restriction, other classes axioms, properties relations, …) when possible afterwards.

### **Implementation choices for specific communities**

* ShapeChange “Map entries” provide a flexible way to choose recommended names for properties and classes. This would enable one to reuse existing specialized classes and properties from external ontologies.
* GeologicUnitView contains mixed information from both GeologicUnit and MappedFeature. A decision must be made to which entity the view must be associated (using the same URI as the GeologicUnit or MappedFeature )

**Meta-model issues (expressivity mismatches between OWL and UML)**

* The placeholder attribute “any” (in GeoSciML Lite) becomes useless property in owl delete it.
* Choice made to replace the “character string” data properties by object properties from GeoSciML Basic, borehole and other ontologies when possible (using the XPath mapping detailed in GeoSciML specification).
* <<type>> and <<FeatureType>> serialise to owl:Class - we need to have further annotation or axiomitisation (e.g. <<datatype>>)
* Abstract class : According to ISO 19150-2, abstract classes in UML are transformed to annotated owl class. But in GSMLsome abstract classes were created to provide an extension point for GSML extension (ex: FoliationAbstractDescription); they provided a bag to list properties. Some might then be revisited/deleted (the only reason to keep them would be for schema mapping purposes but we considered it a low priority use case compared to LinkedOpenData, Websem reasoning)
* The expressiveness of ontology languages should be used to enrich the reasoning: axioms on classes (equivalence, disjointness), and properties relations (inverse, equivalence) and characteristics (transitivity, symmetry, functionality and inverse functionality ).
* UML class union should be transformed using owl:disjointUnionOf
* Rob : The key meta-model issue I see here is the use of a character string (UML option) to hold an IRI in a particular implementation profile - and the trickiness of modelling this as an objectProperty or not. Option could be to model it as an rdfs:Property, and allow implementation profiles to constrain it to an owl:ObjectProperty.

### **Bugs and limitations in software (or things too hard to configure)**

* Association classes must be handled differently: ShapeChange transforms an association class into separate class and properties. Thus, no link is created between the association class and the classes that are initially related by it in the UML. No rule is found in ShapeChange to handle that.

This must be defined afterwards with two properties: associationSource and associationTarget (exactly as in passing from conceptual model to a logical schema). As a solution, this could be locally defined as [association name]+”Source” and [association name]+”Target”. These two properties must have the right domain and range. The direct property between the source and the target automatically created by Shapechange must be deleted.

### **Annotation practices**

* Version the ontology: use owl:priorVersion and owl:versionInfo properties to describe the ontology, and owl:deprecatedClass and owl:deprecatedProperty annotation properties to specify the version status of a class or a property when deprecated.
* Use PROV-O to describe the provenance of the ontology with reference to 19150-2, ShapeChange configuration, ...

### **Proposed behaviour when external classes is specified as properties values**

When a UML class from another schema is referenced (Observation class for example ), it should be replaced by the specialized classes from the ontology of that schema (could be automated in ShapeChange). If such ontology isn’t defined (SWE types for example) use (equivalent) classes from other ontologies or define new ones.

### **Standing issues**

* Usage of skos VS dedicated classes when transforming <<codeList>> from the UML

The pattern proposed by ISO-19150-2 is to create a class for each property designed to hold a “term”. This class shall be a subtype of skos::Concept according to the spec. This is seen as a problem for some as SKOS is not the only possible way to encode vocabularies, as some might prefer to encode vocabularies as formal ontologies.

* Version URI

Do we need to specify where version numbers go in the URI schemes discussed above ?

# **Support material**

## **Configuration references**

Shapechange configuration: <https://github.com/opengeospatial/GeoSciML/blob/master/tools/shapeChange/gsml4_bh.xml>

Example of transforming GeoSciML Borehole UML Model into OWL. Should be re-used for oher (just need to change the source EAP file, appSchemaName, URIbase).

## **GeoSciML encoding example**

Example of transforming GeoSciML Borehole UML Model into OWL

Resulating raw ontology from Shapechange:

<https://github.com/opengeospatial/GeoSciML/blob/master/ontology/1_raw_from_Shapechange/gsmlbh.ttl>

Ontology after manual edition:

<https://github.com/opengeospatial/GeoSciML/blob/master/ontology/2_after_manual_edit/gsmlbh.ttl>

## **HY\_Features encoding example**