**Notes On**

**The Observable Properties Ontology**

In performing an exercise to map IMOS/AODN Parameter Names to the Observable Properties (OP) ontology a number of issues were noted which created uncertainty in how the mappings were to be performed. Whilst it is readily acknowledged that the OP ontology was specifically designed to be a simple ontology that users could easily pick up and apply (which is highly laudable), this simplicity of construction of itself sometimes makes its application uncertain. These brief notes identify where difficulties were found in applying the ontology and in some cases makes suggestions about how these perceived problems might be addressed. Whilst some of these suggestions make the ontology a little more complex, it is felt that they also provide more rigour and may help pave the way for it to be used more easily with reasoning engines in the future.

The main problems encountered during mapping exercises are described below.

**Difficulties Encountered**

*1. Naming (or Labeling) an ‘op:PropertyKind’*

It is not obvious from the ontology element descriptions what constitutes acceptable ‘op:PropertyKind’ instance names. This is significant for an ontology describing observed properties because when referred to in general parlance, observed properties are often couched in terms of the ‘entity’ to which they pertain, or in which they inhere (e.g. ‘Nitrogen Concentration’ or ‘ ‘Temperature of the Water Body’) and sometimes other characterizing information. In the examples just provided the actual quality kinds (or types) of ‘properties’ being referred to are ‘Concentration’ and ‘Temperature’ respectively. ‘Nitrogen’ and ‘Water Body’ provide the entity description for the ‘object’ in which the property inheres. But often, in real-world use the property name (or label) will include more than just the ‘object’ which is being characterized by the property and will also include other contextual qualifying concepts.

The OP provides context for ‘properties’ by qualifying an ‘op:PropertyKind’ using ‘constraints’. These constraints serve to narrow the description of the property being identified. Constraints in OP are OWL ObjectProperties. Identifying and elaborating these ‘constraints’ is the primary function of the ontology. But in order to generate property terms that can be differentiated by their names (as well as by their full ontological descriptions) it is necessary to have articulated some ‘conventions’ regarding which contextualizing constraints can appear in the ‘op:PropertyKind’ instance names. At present this guidance is absent except for an inline comment that the ‘op:PropertyKind’ – “may carry an objectOfInterest property, to point to the substance or taxon with which the property is associated - e.g. tree-height, organism-count, nitrogen-concentration”. It would appear from this text, and examples already provided within eREEFs property descriptions, that the ‘op:PropertyKind’ instance name can contain the object in which it inheres (e.g. tree-height). The text quoted above however, does not actually say that property instance names can carry the ultimate object of interest, it says that ‘properties’ may have a ‘relationship’ with ‘objects of interest’. Indeed they do, but they may also have other ‘constraining relationships’ through ‘op:procedure’, ‘op:featureOfInterest’, and ‘op:matrix’ with other concepts that help to differentiate and characterize the property. The question is - can any, or all of these contextualizing elements legitimately be used in the ‘op:PropertyKind’ instance term name ? Guidance on this would be useful and should reside in the ontology.

There are circumstances where including the Feature Of Interest, as well as an Object Of Interest in a ‘op:PropertyKind’ instance term name is useful for delineating an observed property. Take for example the situation where you would like to use a controlled vocabulary of observed properties in a search interface of a data discovery portal. Observed properties named ‘Wave Height In The Water Body’ and ‘Wave Height In The Troposphere’ permit portal users to improve the precision with which they retrieve datasets matching their property of interest. In this case the differentiation of importance, apart from identifying that ‘height’ pertains to ‘waves’, is the additional description that the ‘waves’ in question are in ‘water’ as opposed to waves propagating in ‘air’. It could be argued that this example could be taken to its logical conclusion by stating that all ‘constraints’ should be included in the term name (which is exactly how the BODC Parameter Names are constructed).

Of course it is possible for datasets, fully annotated with OP ontology elements, to be indexed in search engines using an individually registered base property (i.e. an unconstrained op:PropertyKind) along with each of its ‘op:PropertyKind’ constraints and then search interfaces could be customized to permit faceted searching over these base properties and their individual constraints. This could be achieved using some sort of cascading menu that gradually narrows (i.e. further qualifies) base properties until the user has specified the desired level of description (through gradually aggregating any identified OP constraints) before hitting the search button.

There are obviously arguments both for and against constraint-inclusive observable property instance names. Either way some guidance on the intended naming conventions within the ontology would be useful in terms of helping to determine the appropriate level of detail that should be captured as part of the op:PropertyKind term name.

*2. op:objectOfInterest and op:SusbtanceOrTaxon*

At the moment the entity in which a property can inhere seems to be confined to ‘op:Substance’ and ‘op:Taxon’. These classes are sub-classes of the superclass - ‘op:’SubstanceOrTaxon’ (see Figure 1). The text definition for this superclass is “Class of stuff and things, individuals of which identify a class of stuff and things that make observed properties concrete.” The sub-classes (i.e., ‘op:Substance’ and ‘op:Taxon’) have no textual definition and the ontology places almost no restrictions on them, so from an ontological perspective they remain basically undefined.

The first problem perceived here relates to the issue that the classes used above (as defined) don’t appear to cover all of the types of entities in which an observable property might inhere. The second problem simply relates to the lack of rigour used to describe the classes.

In the (observed property) example provided earlier – i.e., ‘Wave Height In The Water Body’, the entity - ‘Wave’ cannot strictly be considered a ‘substance’ or a ‘taxon’, in the sense that these terms are generally used in natural language. Since there is no elaboration within OP on what these sub-classes of thing should be – only the general usage sense of the words can be assumed. A ‘Wave’ is really more like a type of ‘physical process’ or perhaps ‘phenomenon’ – not a ‘substance’. At the risk of complicating the ontology it might be better to expand on the various categories of thing in which properties can inhere (and as a minimum give them a textual definition) so that entities such as ‘Wave’ can be more comfortably accommodated (see Figure 2).

A further approach could be to consider the range of an ‘op:objectOfInterest’ property to be an ‘op:ObjectOfInterestRole’ class instead of an ‘op:SubstanceOrTaxon’ class (see Figure 3).

An ‘op:ObjectOfInterestRole’ class is a concept that identifies a ‘role’ played by another concept (e.g. ‘Teacher’ is an instance of ‘TeacherRole’ and could be played by a concept ‘Person’ or and individual (John Smith) of the class ‘Person’). In the case of OP - ‘op:ObjectOfInterestRole’ could be played by a range of different, but existing concepts drawn from other ontologies (e.g. SWEET). The range of op:featureOfInterest could similarly be couched as a ‘role’ since at present the range of op:featureOfInterest has the same restriction as that of its parent property (i.e.,‘Feature:AnyFeature’) which currently also seems under-specified from the standpoint of the OP ontology. Figure 4 takes this idea of ‘roles’ to the extreme and channels all ‘constraints’ through a generic ‘ConstraintOfInterest” role concept.

In an ontology, particularly if there is any intent to use it in the future for reasoning, it is wise to make sure that classes of thing don’t end up ‘tangled’. Tangled ontologies have subsumption hierarchies with more than one superclass per class. I’m concerned that there is the potential for entanglement in OP descriptions that include classes of thing that may be simultaneously considered a sub-class of ‘om:Feature Type’ and also a sub-class of ‘op:SubstanceOrTaxon’. This is because it is conceivable that the concept - ‘Water Body’ for example can be viewed in different individual contexts to be both an ‘Object Of Interest’ (and thus a sub-class, or individual of ‘op:SusbtanceOrTaxon’) and a Feature Of Interest (thus a sub-class, or individual of ‘Feature Type’) as in ’Temperature of The Water Body’ where ‘Water Body’ is the object in which temperature inheres vs ‘Wave Height In The Water Body’ where ‘Water Body’ is not the object, but the feature of interest.

*3. op:matrix*

The OWL Object Property ‘op:matrix’ is defined as a sub-property of op:featureOfInterest. It has no specifically declared domain and range restrictions. The textual definition provided says: “In an observable property definition, the matrix is a special case of a feature-of-interest that provides the context (container feature or medium) for an observable property”. There is not enough information in the text definition provided, or in the ontology rules for a user (e.g. me) to understand what is intended here.

*4. qudt ontology*

In many cases our (AODN) Parameter Names contain propertykinds that are a combination of a quantity and a mathematical operator (like ‘Average Height’). Is it OK to simply add these terms with mathematical qualifiers to existing QUDT or EREEFs QualityKind descriptions (which may already include the base quantity) and just treat them as new un-related concepts since I think QUDT is primarily comprised of instances rather than classes ?

*5. ScaledQuantityKind*

By definition a ScaledQuantityKind must carry a Unit Of Measure. If the number is just a ‘count’ (e.g. number of fish) is it then acceptable to use say - ‘individuals’ as the unit of measure ?

IsIndividualOf

**Figure 1 Existing OP Ontology**

e.g. individual

e.g. sweet:EarthRealm

op:matrix

op:featureOfInterest

Feature:AnyFeature

Feature:AnyFeature

Is a

op:Taxon

op:Substance

Is a

op:procedure

OM:Process

has

op:SubstanceOrTaxon

op:objectOfInterest

Is a

qudt:Unit

Is a

op:ScaledQuantityKind

op:QualityKind

op:PropertyKind

**Figure 2 Modified OP Ontology (‘op:Entity’ could be a single superclass for the various sub-classes, or be defined as the ‘union of’ the different classes shown).**

IndividualOf

individual

op:NonBioticPhysicalObject

Is a

op:BioticPhysicalObject

Is a

Is a

op:PhysicalPhenomenon

op:Substance

op:Entity

op:matrix

op:featureOfInterest

Feature:AnyFeature

Feature:AnyFeature

Is a

op:procedure

OM:Process

has

op:objectOfInterest

Is a

qudt:Unit

Is a

op:ScaledQuantityKind

op:QualityKind

op:PropertyKind

**Figure 3 Possible Modified Structure For the OP Ontology using a Role Concept**

playsRoleOf

playsRoleOf

playsRoleOf

playsRoleOf

playsRoleOf

sweet:PhysicalPhenomena

sweet:LivingSubstance

playsRoleOf

playsRoleOf

op:procedure

op:Procedure

sweet:PhysicalProcess

sweet:EarthRealm

sweet:NonLivingSubstance

has

op:featureOfInterest

Is a

Is a

op:ObjectOfInterest

op:FeatureOfInterest

op:objectOfInterest

Is a

qudt:Unit

op:Role

Is a

op:ScaledQuantityKind

op:QualityKind

op:PropertyKind

**Figure 4 Possible Modified Structure For the OP Ontology (using Role concept and a generic ‘ConstraintOfInterest’ Class)**

playsRoleOf

OM:Process

playsRoleOf

sweet:PhysicalPhenomena

ObjectOfInterest

Procedure

Matrix

FeatureOfInterest

has

qudt:Unit

playsRoleOf

Is a

sweet:PhysicalProcess

playsRoleOf

sweet:LivingSubstance

playsRoleOf

sweet:NonLivingSubstance

playsRoleOf

sweet:EarthRealm

op:ConstraintOfInterest

op:Role

op:constraint

Is a

Is a

op:ScaledQuantityKind

op:QualityKind

op:PropertyKind