5 Data content and structure

This data specification defines the following application schemas:
- The Oceanographic Geographical Features application schema.

5.1 Application schemas – Overview

5.1.1 Application schemas included in the IRs

Articles 3, 4 and 5 of the Implementing Rules lay down the requirements for the content and structure of the data sets related to the INSPIRE Annex themes.

IR Requirement
Article 4
Types for the Exchange and Classification of Spatial Objects

1. For the exchange and classification of spatial objects from data sets meeting the conditions laid down in Article 4 of Directive 2007/2/EC, Member States shall use the spatial object types and associated data types, enumerations and code lists that are defined in Annexes II, III and IV for the themes the data sets relate to.

2. Spatial object types and data types shall comply with the definitions and constraints and include the attributes and association roles set out in the Annexes.

3. The enumerations and code lists used in attributes or association roles of spatial object types or data types shall comply with the definitions and include the values set out in Annex II. The enumeration and code list values are uniquely identified by language-neutral mnemonic codes for computers. The values may also include a language-specific name to be used for human interaction.

The types to be used for the exchange and classification of spatial objects from data sets related to the spatial data theme Oceanographic geographical features are defined in the following application schemas (see sections 5.3):

- The Oceanographic Geographical Features application schema (section 5.3).

The application schemas specify requirements on the properties of each spatial object including its multiplicity, domain of valid values, constraints, etc.

NOTE The application schemas presented in this section contain some additional information that is not included in the Implementing Rules, in particular multiplicities of attributes and association roles.

TG Requirement 1	Spatial object types and data types shall comply with the multiplicities defined for the attributes and association roles in this section.

An application schema may include references (e.g. in attributes or inheritance relationships) to common types or types defined in other spatial data themes. These types can be found in a sub-section called “Imported Types” at the end of each application schema section. The common types referred to from application schemas included in the IRs are addressed in Article 3.

IR Requirement
Article 3
Common Types

Types that are common to several of the themes listed in Annexes I, II and III to Directive 2007/2/EC shall conform to the definitions and constraints and include the attributes and association roles set out in Annex I.

NOTE Since the IRs contain the types for all INSPIRE spatial data themes in one document, Article 3 does not explicitly refer to types defined in other spatial data themes, but only to types defined in external data models.

Common types are described in detail in the Generic Conceptual Model [DS-D2.7], in the relevant international standards (e.g. of the ISO 19100 series) or in the documents on the common INSPIRE models [DS-D2.10.x]. For detailed descriptions of types defined in other spatial data themes, see the corresponding Data Specification TG document [DS-D2.8.x].

5.1.2 Additional recommended application schemas

There is no additional application schemas defined for the theme *Oceanographic geographical features.*

5.2 Basic notions

This section explains some of the basic notions used in the INSPIRE application schemas. These explanations are based on the GCM [DS-D2.5].

5.2.1 Notation

5.2.1.1. Unified Modeling Language (UML)

The application schemas included in this section are specified in UML, version 2.1. The spatial object types, their properties and associated types are shown in UML class diagrams.

NOTE For an overview of the UML notation, see Annex D in [ISO 19103].

The use of a common conceptual schema language (i.e. UML) allows for an automated processing of application schemas and the encoding, querying and updating of data based on the application schema – across different themes and different levels of detail.

The following important rules related to class inheritance and abstract classes are included in the IRs.

IR Requirement
Article 5
Types

(…)

2. Types that are a sub-type of another type shall also include all this type's attributes and association roles.

3. Abstract types shall not be instantiated.

The use of UML conforms to ISO 19109 8.3 and ISO/TS 19103 with the exception that UML 2.1 instead of ISO/IEC 19501 is being used. The use of UML also conforms to ISO 19136 E.2.1.1.1-E.2.1.1.4.

NOTE ISO/TS 19103 and ISO 19109 specify a profile of UML to be used in conjunction with the ISO 19100 series. This includes in particular a list of stereotypes and basic types to be used in application schemas. ISO 19136 specifies a more restricted UML profile that allows for a direct encoding in XML Schema for data transfer purposes.

To model constraints on the spatial object types and their properties, in particular to express data/data set consistency rules, OCL (Object Constraint Language) is used as described in ISO/TS 19103, whenever possible. In addition, all constraints are described in the feature catalogue in English, too.

NOTE Since "void" is not a concept supported by OCL, OCL constraints cannot include expressions to test whether a value is a *void* value. Such constraints may only be expressed in natural language.

5.2.1.2. Stereotypes

In the application schemas in this section several stereotypes are used that have been defined as part of a UML profile for use in INSPIRE [DS-D2.5]. These are explained in Table 1 below.

Table 1 – Stereotypes (adapted from [DS-D2.5])

Stereotype	Model element	Description
applicationSchema	Package	An INSPIRE application schema according to ISO 19109 and the Generic Conceptual Model.
leaf	Package	A package that is not an application schema and contains no packages.
featureType	Class	A spatial object type.
type	Class	A type that is not directly instantiable, but is used as an abstract collection of operation, attribute and relation signatures. This stereotype should usually not be used in INSPIRE application schemas as these are on a different conceptual level than classifiers with this stereotype.
dataType	Class	A structured data type without identity.
union	Class	A structured data type without identity where exactly one of the properties of the type is present in any instance.
enumeration	Class	An enumeration.
codeList	Class	A code list.
import	Dependency	The model elements of the supplier package are imported.
voidable	Attribute, association role	A voidable attribute or association role (see section 5.2.2).
lifeCycleInfo	Attribute, association role	If in an application schema a property is considered to be part of the life-cycle information of a spatial object type, the property shall receive this stereotype.
version	Association role	If in an application schema an association role ends at a spatial object type, this stereotype denotes that the value of the property is meant to be a specific version of the spatial object, not the spatial object in general.

5.2.2 Voidable characteristics

The «voidable» stereotype is used to characterise those properties of a spatial object that may not be present in some spatial data sets, even though they may be present or applicable in the real world. This does *not* mean that it is optional to provide a value for those properties.

For all properties defined for a spatial object, a value has to be provided – either the corresponding value (if available in the data set maintained by the data provider) or the value of *void.* A *void* value shall imply that no corresponding value is contained in the source spatial data set maintained by the data provider or no corresponding value can be derived from existing values at reasonable costs.

Recommendation 1	The reason for a *void* value should be provided where possible using a listed value from the VoidReasonValue code list to indicate the reason for the missing value.

The VoidReasonValue type is a code list, which includes the following pre-defined values:

– *Unpopulated*: The property is not part of the dataset maintained by the data provider. However, the characteristic may exist in the real world. For example when the "elevation of the water body above the sea level" has not been included in a dataset containing lake spatial objects, then the reason for a void value of this property would be 'Unpopulated'. The property receives this value for all spatial objects in the spatial data set.
– *Unknown*: The correct value for the specific spatial object is not known to, and not computable by the data provider. However, a correct value may exist. For example when the "elevation of the water body above the sea level" *of a certain lake* has not been measured, then the reason for a void value of this property would be 'Unknown'. This value is applied only to those spatial objects where the property in question is not known.
– *Withheld*: The characteristic may exist, but is confidential and not divulged by the data provider.

NOTE It is possible that additional reasons will be identified in the future, in particular to support reasons / special values in coverage ranges.

The «voidable» stereotype does not give any information on whether or not a characteristic exists in the real world. This is expressed using the multiplicity:

– If a characteristic may or may not exist in the real world, its minimum cardinality shall be defined as 0. For example, if an Address may or may not have a house number, the multiplicity of the corresponding property shall be 0..1.
– If at least one value for a certain characteristic exists in the real world, the minimum cardinality shall be defined as 1. For example, if an Administrative Unit always has at least one name, the multiplicity of the corresponding property shall be 1..*.

In both cases, the «voidable» stereotype can be applied. In cases where the minimum multiplicity is 0, the absence of a value indicates that it is known that no value exists, whereas a value of void indicates that it is not known whether a value exists or not.

EXAMPLE If an address does not have a house number, the corresponding Address object should not have any value for the «voidable» attribute house number. If the house number is simply not known or not populated in the data set, the Address object should receive a value of *void* (with the corresponding void reason) for the house number attribute.

5.2.3 Enumerations

Enumerations are modelled as classes in the application schemas. Their values are modelled as attributes of the enumeration class using the following modelling style:

– No initial value, but only the attribute name part, is used.
– The attribute name conforms to the rules for attributes names, i.e. is a lowerCamelCase name. Exceptions are words that consist of all uppercase letters (acronyms).

IR Requirement
Article 6
Code Lists and Enumerations

(…)

5) Attributes or association roles of spatial object types or data types that have an enumeration type may only take values from the lists specified for the enumeration type."

5.2.4 Code lists

Code lists are modelled as classes in the application schemas. Their values, however, are managed outside of the application schema.

5.2.4.1. Code list types

The IRs distinguish the following types of code lists.

IR Requirement
Article 6
Code Lists and Enumerations

1) Code lists shall be of one of the following types, as specified in the Annexes:
   a) code lists whose allowed values comprise only the values specified in this Regulation;
   b) code lists whose allowed values comprise the values specified in this Regulation and narrower values defined by data providers;
   c) code lists whose allowed values comprise the values specified in this Regulation and additional values at any level defined by data providers;
   d) code lists, whose allowed values comprise any values defined by data providers.

   For the purposes of points (b), (c) and (d), in addition to the allowed values, data providers may use the values specified in the relevant INSPIRE Technical Guidance document available on the INSPIRE web site of the Joint Research Centre.

The type of code list is represented in the UML model through the tagged value *extensibility*, which can take the following values:

- *none*, representing code lists whose allowed values comprise only the values specified in the IRs (type a);
- *narrower*, representing code lists whose allowed values comprise the values specified in the IRs and narrower values defined by data providers (type b);
- *open*, representing code lists whose allowed values comprise the values specified in the IRs and additional values at any level defined by data providers (type c); and
- *any*, representing code lists, for which the IRs do not specify any allowed values, i.e. whose allowed values comprise any values defined by data providers (type d).

Recommendation 2 Additional values defined by data providers should not replace or redefine any value already specified in the IRs.

NOTE This data specification may specify recommended values for some of the code lists of type (b), (c) and (d) (see section 5.2.4.3). These recommended values are specified in a dedicated Annex.

In addition, code lists can be hierarchical, as explained in Article 6(2) of the IRs.

IR Requirement
Article 6
Code Lists and Enumerations

(…)

2) Code lists may be hierarchical. Values of hierarchical code lists may have a more generic parent value. Where the valid values of a hierarchical code list are specified in a table in this Regulation, the parent values are listed in the last column.

The type of code list and whether it is hierarchical or not is also indicated in the feature catalogues.

5.2.4.2. Obligations on data providers

IR Requirement
Article 6
Code Lists and Enumerations

(….)

3) Where, for an attribute whose type is a code list as referred to in points (b), (c) or (d) of paragraph 1, a data provider provides a value that is not specified in this Regulation, that value and its definition shall be made available in a register.

4) Attributes or association roles of spatial object types or data types whose type is a code list may only take values that are allowed according to the specification of the code list.

Article 6(4) obliges data providers to use only values that are allowed according to the specification of the code list. The “allowed values according to the specification of the code list” are the values explicitly defined in the IRs plus (in the case of code lists of type (b), (c) and (d)) additional values defined by data providers.

For attributes whose type is a code list of type (b), (c) or (d) data providers may use additional values that are not defined in the IRs. Article 6(3) requires that such additional values and their definition be made available in a register. This enables users of the data to look up the meaning of the additional values used in a data set, and also facilitates the re-use of additional values by other data providers (potentially across Member States).

NOTE Guidelines for setting up registers for additional values and how to register additional values in these registers is still an open discussion point between Member States and the Commission.

5.2.4.3. Recommended code list values

For code lists of type (b), (c) and (d), this data specification may propose additional values as a recommendation (in a dedicated Annex). These values will be included in the INSPIRE code list register. This will facilitate and encourage the usage of the recommended values by data providers since the obligation to make additional values defined by data providers available in a register (see section 5.2.4.2) is already met.

Recommendation 3 Where these Technical Guidelines recommend values for a code list in addition to those specified in the IRs, these values should be used.

NOTE For some code lists of type (d), no values may be specified in these Technical Guidelines. In these cases, any additional value defined by data providers may be used.

5.2.4.4. Governance

The following two types of code lists are distinguished in INSPIRE:

- *Code lists that are governed by INSPIRE (INSPIRE-governed code lists).* These code lists will be managed centrally in the INSPIRE code list register. Change requests to these code lists (e.g. to add, deprecate or supersede values) are processed and decided upon using the INSPIRE code list register's maintenance workflows.

  INSPIRE-governed code lists will be made available in the INSPIRE code list register at *http://inspire.ec.europa.eu/codelist/<CodeListName>*. They will be available in SKOS/RDF, XML and HTML. The maintenance will follow the procedures defined in ISO 19135. This means that the only allowed changes to a code list are the addition, deprecation or supersession of values, i.e. no value will ever be deleted, but only receive different statuses (valid, deprecated, superseded). Identifiers for values of INSPIRE-governed code lists are constructed using the pattern *http://inspire.ec.europa.eu/codelist/<CodeListName>*/<value>.

- *Code lists that are governed by an organisation outside of INSPIRE (externally governed code lists).* These code lists are managed by an organisation outside of INSPIRE, e.g. the World Meteorological Organization (WMO) or the World Health Organization (WHO). Change requests to these code lists follow the maintenance workflows defined by the maintaining organisations. Note that in some cases, no such workflows may be formally defined.

  Since the updates of externally governed code lists is outside the control of INSPIRE, the IRs and these Technical Guidelines reference a specific version for such code lists.

  The tables describing externally governed code lists in this section contain the following columns:
  - The *Governance* column describes the external organisation that is responsible for maintaining the code list.
  - The *Source* column specifies a citation for the authoritative source for the values of the code list. For code lists, whose values are mandated in the IRs, this citation should include the version of the code list used in INSPIRE. The version can be specified using a version number or the publication date. For code list values recommended in these Technical Guidelines, the citation may refer to the "latest available version".
  - In some cases, for INSPIRE only a subset of an externally governed code list is relevant. The subset is specified using the *Subset* column.
  - The *Availability* column specifies from where (e.g. URL) the values of the externally governed code list are available, and in which formats. Formats can include machine-readable (e.g. SKOS/RDF, XML) or human-readable (e.g. HTML, PDF) ones.

  Code list values are encoded using http URIs and labels. Rules for generating these URIs and labels are specified in a separate table.

Recommendation 4	The http URIs and labels used for encoding code list values should be taken from the INSPIRE code list registry for INSPIRE-governed code lists and generated according to the relevant rules specified for externally governed code lists.

NOTE Where practicable, the INSPIRE code list register could also provide http URIs and labels for externally governed code lists.

5.2.4.5. Vocabulary

For each code list, a tagged value called "vocabulary" is specified to define a URI identifying the values of the code list. For INSPIRE-governed code lists and externally governed code lists that do not have a persistent identifier, the URI is constructed following the pattern *http://inspire.ec.europa.eu/codelist/<UpperCamelCaseName>*.

If the value is missing or empty, this indicates an empty code list. If no sub-classes are defined for this empty code list, this means that any code list may be used that meets the given definition.

An empty code list may also be used as a super-class for a number of specific code lists whose values may be used to specify the attribute value. If the sub-classes specified in the model represent all valid extensions to the empty code list, the subtyping relationship is qualified with the standard UML constraint "{complete,disjoint}".

5.2.5 Identifier management

IR Requirement
Article 9
Identifier Management

1. The data type Identifier defined in Section 2.1 of Annex I shall be used as a type for the external object identifier of a spatial object.
2. The external object identifier for the unique identification of spatial objects shall not be changed during the life-cycle of a spatial object.

NOTE 1 An external object identifier is a unique object identifier which is published by the responsible body, which may be used by external applications to reference the spatial object. [DS-D2.5]

NOTE 2 Article 9(1) is implemented in each application schema by including the attribute *inspireId* of type Identifier.

NOTE 3 Article 9(2) is ensured if the *namespace* and *localId* attributes of the Identifier remains the same for different versions of a spatial object; the *version* attribute can of course change.

5.2.6 Geometry representation

IR Requirement
Article 12
Other Requirements & Rules

1. The value domain of spatial properties defined in this Regulation shall be restricted to the Simple Feature spatial schema as defined in Herring, John R. (ed.), OpenGIS® Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture, version 1.2.1, Open Geospatial Consortium, 2011, unless specified otherwise for a specific spatial data theme or type.

NOTE 1 The specification restricts the spatial schema to 0-, 1-, 2-, and 2.5-dimensional geometries where all curve interpolations are linear and surface interpolations are performed by triangles.

NOTE 2 The topological relations of two spatial objects based on their specific geometry and topology properties can in principle be investigated by invoking the operations of the types defined in ISO 19107 (or the methods specified in EN ISO 19125-1).

5.2.7 Temporality representation

The application schema(s) use(s) the derived attributes "beginLifespanVersion" and "endLifespanVersion" to record the lifespan of a spatial object.

The attributes "beginLifespanVersion" specifies the date and time at which this version of the spatial object was inserted or changed in the spatial data set. The attribute "endLifespanVersion" specifies the date and time at which this version of the spatial object was superseded or retired in the spatial data set.

NOTE 1 The attributes specify the beginning of the lifespan of the version in the spatial data set itself, which is different from the temporal characteristics of the real-world phenomenon described by the spatial object. This lifespan information, if available, supports mainly two requirements: First, knowledge about the spatial data set content at a specific time; second, knowledge about changes to a data set in a specific time frame. The lifespan information should be as detailed as in the data set (i.e., if the lifespan information in the data set includes seconds, the seconds should be represented in data published in INSPIRE) and include time zone information.

NOTE 2 Changes to the attribute "endLifespanVersion" does not trigger a change in the attribute "beginLifespanVersion".

IR Requirement
Article 10
Life-cycle of Spatial Objects

(…)

3. Where the attributes beginLifespanVersion and endLifespanVersion are used, the value of endLifespanVersion shall not be before the value of beginLifespanVersion.

NOTE The requirement expressed in the IR Requirement above will be included as constraints in the UML data models of all themes.

Recommendation 5 If life-cycle information is not maintained as part of the spatial data set, all spatial objects belonging to this data set should provide a void value with a reason of "unpopulated".

5.2.7.1. Validity of the real-world phenomena

The application schema(s) use(s) the attributes "validFrom" and "validTo" to record the validity of the real-world phenomenon represented by a spatial object.

The attributes "validFrom" specifies the date and time at which the real-world phenomenon became valid in the real world. The attribute "validTo" specifies the date and time at which the real-world phenomenon is no longer valid in the real world.

Specific application schemas may give examples what “being valid” means for a specific real-world phenomenon represented by a spatial object.

IR Requirement
Article 12
Other Requirements & Rules

(…)

3. Where the attributes validFrom and validTo are used, the value of validTo shall not be before the value of validFrom.

NOTE The requirement expressed in the IR Requirement above will be included as constraints in the UML data models of all themes.

5.2.8 Coverages

Coverage functions are used to describe characteristics of real-world phenomena that vary over space and/or time. Typical examples are temperature, elevation, precipitation, imagery. A coverage contains a set of such values, each associated with one of the elements in a spatial, temporal or spatio-temporal domain. Typical spatial domains are point sets (e.g. sensor locations), curve sets (e.g. isolines), grids (e.g. orthoimages, elevation models), etc.

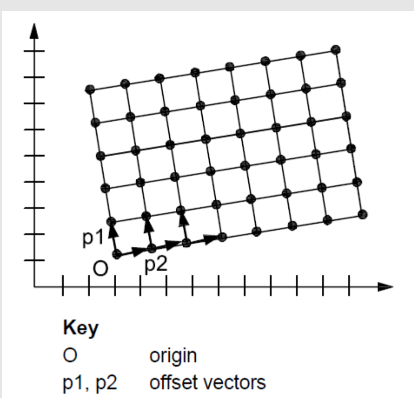
In INSPIRE application schemas, coverage functions are defined as properties of spatial object types where the type of the property value is a realisation of one of the types specified in ISO 19123.

To improve alignment with coverage standards on the implementation level (e.g. ISO 19136 and the OGC Web Coverage Service) and to improve the cross-theme harmonisation on the use of coverages in INSPIRE, an application schema for coverage types is included in the Generic Conceptual Model in 9.9.4. This application schema contains the following coverage types:

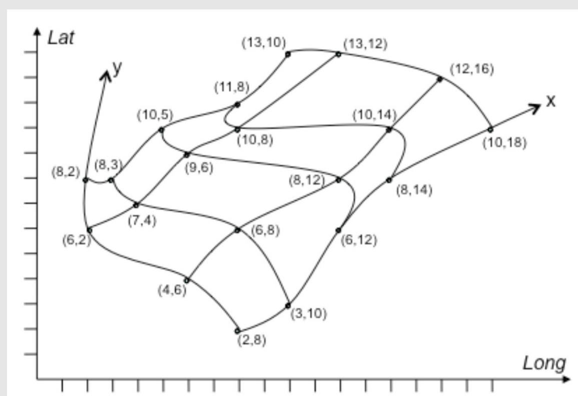
- *RectifiedGridCoverage*: coverage whose domain consists of a rectified grid – a grid for which there is an affine transformation between the grid coordinates and the coordinates of a coordinate reference system (see Figure 2, left).
- *ReferenceableGridCoverage*: coverage whose domain consists of a referenceable grid – a grid associated with a transformation that can be used to convert grid coordinate values to values of coordinates referenced to a coordinate reference system (see Figure 2, right).

In addition, some themes make reference to the types TimeValuePair and Timeseries defined in Taylor, Peter (ed.), *OGC® WaterML 2.0: Part 1 – Timeseries, v2.0.0,* Open Geospatial Consortium, 2012. These provide a representation of the time instant/value pairs, i.e. time series (see Figure 3).

Where possible, only these coverage types (or a subtype thereof) are used in INSPIRE application schemas.



(Source: ISO 19136:2007)



(Source: GML 3.3.0)

Figure 2 – Examples of a rectified grid (left) and a referenceable grid (right)

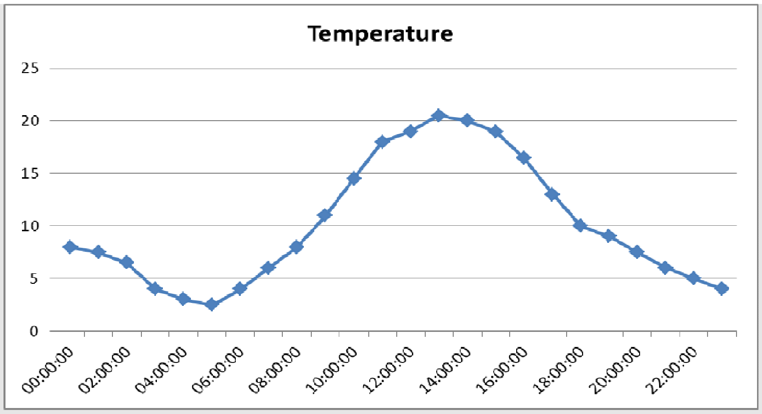


Figure 3 – Example of a time series

5.3 Application schema Oceanographic Geographical Features

5.3.1 Description

5.3.1.1. Narrative description

An Oceanographic Geographical Feature (abbreviated to "Ocean Feature" or "OF") describes the physical and chemical phenomena of a sea region (known as '*SeaArea*' in the Inspire Sea Regions model).

5.3.1.2. UML Overview

Some examples of Ocean Features are:

- A time series of measurements of water level from a tide gauge
- A satellite gridded field of ocean colour
- A one off sea surface temperature measurement made by hand with a thermometer
- An ocean climate model predicting future changes of salinity over time on a model grid.
- Ex-situ measurement of suspended sediment concentration
- In situ measurements or ex situ analysed collected samples gathered from vessels visiting a specified location

In each of these cases some estimation of the value of a property (water level, ocean colour, temperature, salinity) is made using some procedure. For the OF theme we directly build upon the ISO 19156 Observations and Measurements (O&M) specification which provides a framework to describe the relationships between the observation (or simulation) event, the observed property, the procedure used, the observation result and the feature of interest.

ISO 19156 states that:

> *An observation is an act associated with a discrete time instant or period through which a number, term or other symbol is assigned to a phenomenon. It involves application of a specified **procedure**, such as a sensor, instrument, algorithm or process chain. The procedure may be applied in-situ, remotely, or ex-situ with respect to **the sampling location**. The **result** of an observation is an estimate of the **value** of a **property** of some **feature**.* [ISO 19156]

Furthermore:

> *The key idea is that the observation result is an estimate of the value of some property of the feature of interest, and the other observation properties provide context or metadata to support evaluation, interpretation and use of the result.* [ISO 19156]

By building on the ISO 19156 standard model it is expected that interoperability between domains will be increased. Non-specialist software which understands ISO 19156 will be able to interrogate Ocean Features and at least identify the feature of interest, the observation time, the location etc.

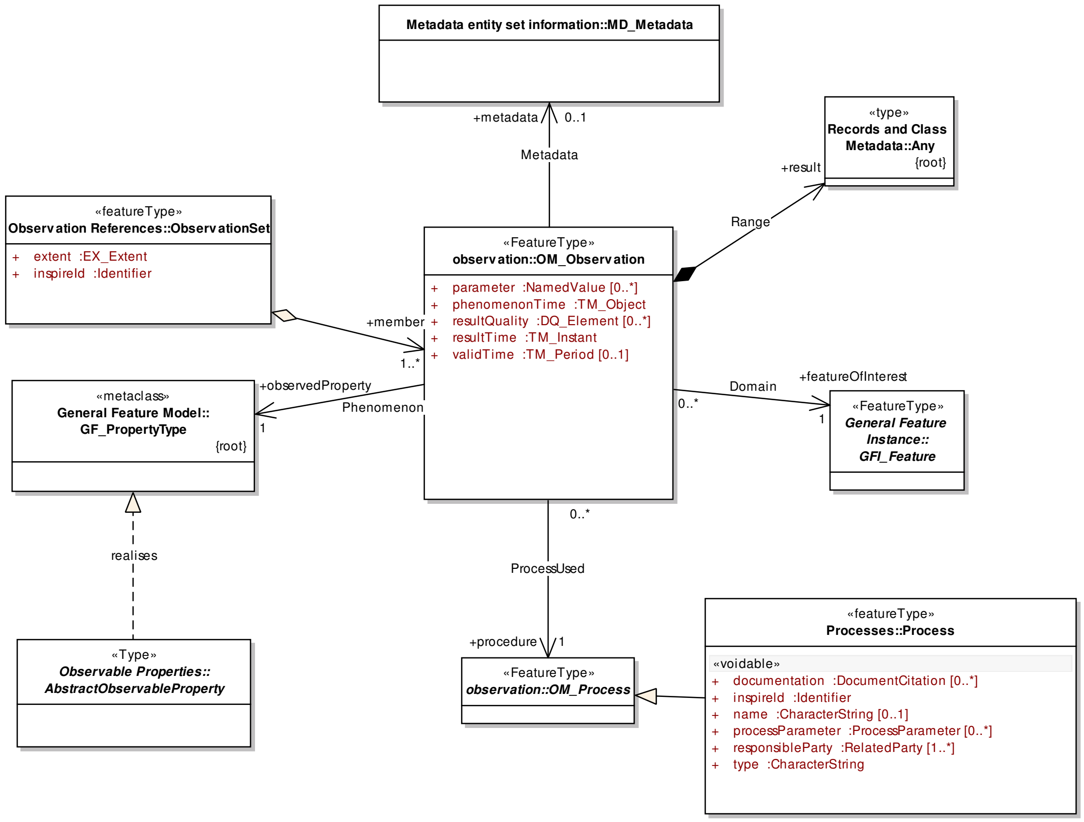


Figure 4 Overview of ISO 19156 OM_Observation

ISO 19156 is a very generic model which may be applied in many different thematic domains. For the purposes of interoperability in INSPIRE we have profiled the O&M model to add further precision about the types of processes, observable properties and features of interest that are used.

It is recognised that the types of data structure that are common means of capturing and presenting information about the oceans such as point observations, profiles, gridded data etc are also common when monitoring weather or climate in the atmosphere. Therefore the OF and AC/MF themes have jointly developed a set of core spatial object types in INSPIRE, which can be found in the Generic Conceptual Model and are more fully documented in document D2.9 (O&M in INSPIRE).

The key aspects of the shared model documented in D2.9 that are relevant to OF are:

- Specialised Observation types – GridObservations, PointObservations etc.
- An ObservationSet type, plus one specialisation PointObservationCollection
- An Process type
- A model for describing Observable Properties (temperature, salinity etc).
- A mechanism for linking between Environmental Monitoring Facilities and Observations

The OF model and schema itself is therefore minimal (since most of the feature types are in the shared Observations model). The OF schema itself only contains information about particular vocabularies (codelists) which must be used when describing OF data.

5.3.1.2.1. Specialised Observations, as used in OF; Overview

The 'Specialised Observations' package in the Generic Conceptual Model contains seven specialisations of OM_Observation, (along with a specialisation of ObservationSet discussed in section 5.3.1.2.2).

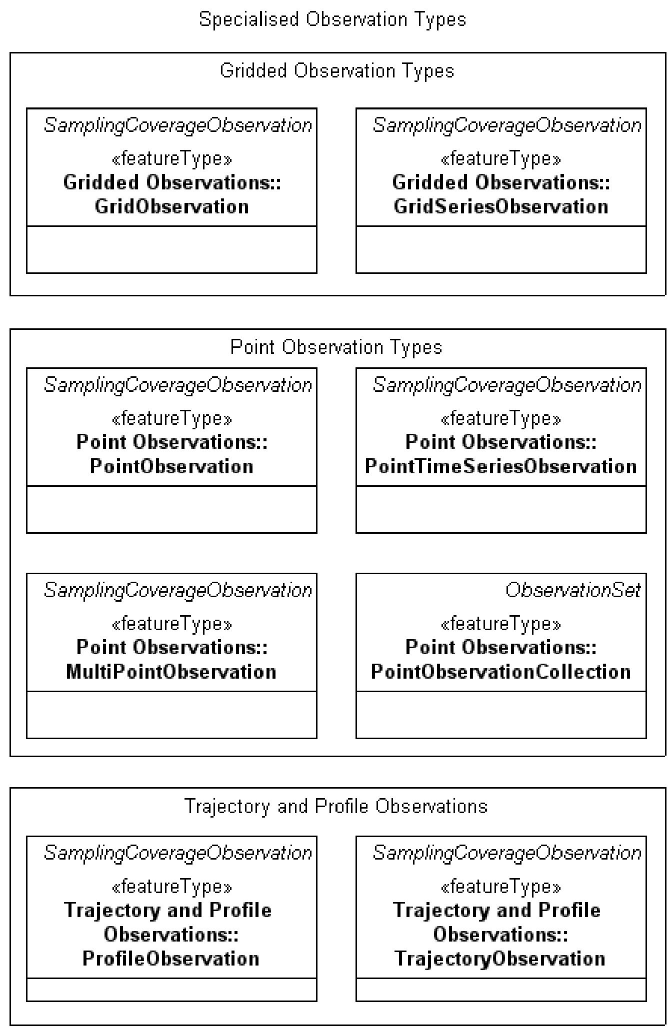


Figure 5 Specialised Observation Types from Generic Conceptual Model (see also document D2.9)

For OF, the 'core' types used are:

- PointObservation
- PointTimeSeriesObservation
- MultiPointObservation
- GridObservation
- GridSeriesObservation
- PointObservationCollection

IR Requirement
Annex IV, Section 14.3

Theme-specific Requirements

Data related to the theme Oceanographic Geographical Features shall be made available using the following types defined in the Specialised Observations package in Annex I: PointObservation, PointTimeSeriesObservation, MultiPointObservation, GridObservation, GridSeriesObservation, PointObservationCollection.

The other types may be used to represent oceanographic data and are provided for convenience and consistency with the core types but these types are not viewed as being central to the INSPIRE OF theme. These types are:

- ProfileObservation
- TrajectoryObservation

Recommendation 6 The following types from the GCM may be used to describe OF data: ProfileObservation, TrajectoryObservation

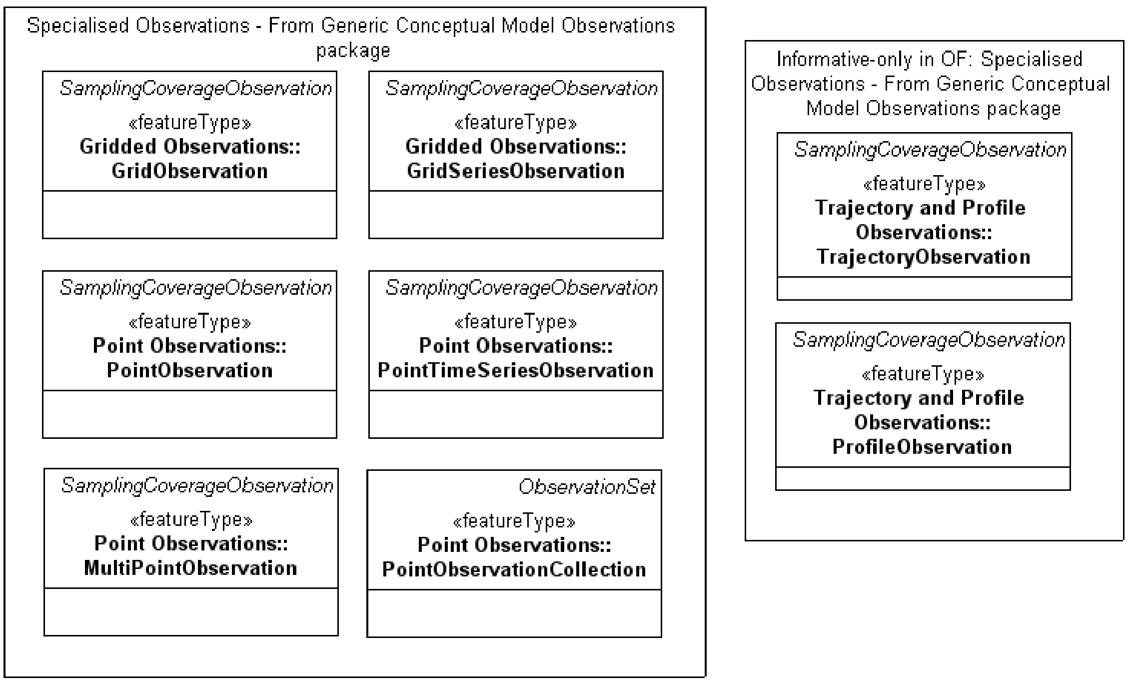


Figure 6 Distinction between normative and informative feature types used in OF

All the specialised Observation types essentially add 'constraints' to the underlying O&M model which characterise the result of the observation and the sampling regime used[1]. For example, a PointTimeSeriesObservation is a timeseries at a single point in space (e.g. at a fixed station), so the 'Spatial Sampling Feature' in 19156 must be a spatial sampling point, and the 'phenomenonTime' must be a time period i.e. the observation must be taken over a period of time. The type of the result must be a set of time, value pairs. Illustrated examples and further descriptions of these types are given in D2.9.

In actual fact, the specialised Observation types do not specialise OM_Observation directly but specialise the informative O&M class *SpecialisedCoverageObservation*, which in turn specialises *DiscreteCoverageObservation*. These two classes between them ensure that the result of the

[1] This pattern was modelled on the approach taken in Climate Science Modelling Language version 3 (OGC Pending Docs 11_021) which extends ISO 19156.

observation is a Coverage, and the feature of interest is a ‘Spatial Sampling Feature’ e.g. a point, an area, a line.

This pattern is consistent with the acquisition of ocean measurements; when taking ocean measurements, we are interested in some property of the ocean, therefore the ultimate feature of interest is the ocean or sea but it is impossible to measure the entire ocean. Therefore measurements are always made against some sampling proxy, such as a set of points, or a grid. O&M refers to this proxy feature as a Spatial Sampling Feature. The result of the measurement is a coverage – a set of values for locations within in the sampling proxy. This is why all OF observations are logically specialisms of *SamplingCoverageObservation*.

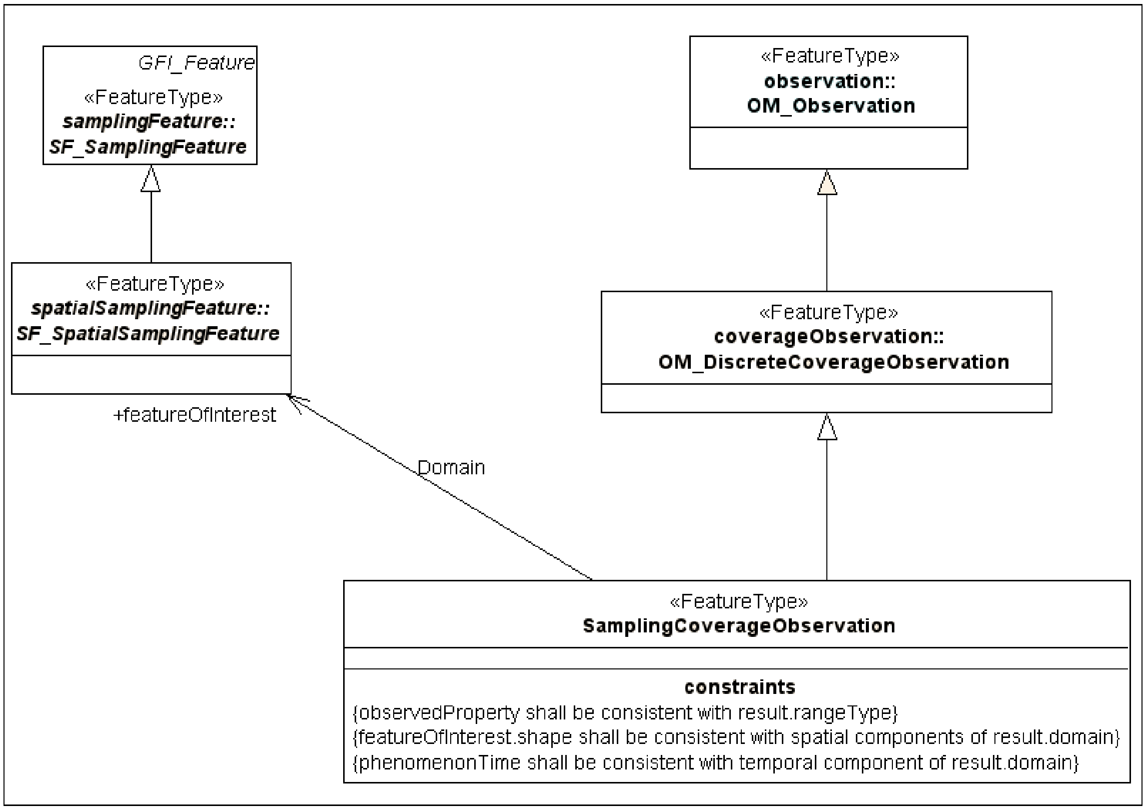


Figure 7 Overview of SamplingCoverageObservation

Note that since these *SamplingCoverageObservation* and *DiscreteCoverageObservation* classes are not realised in the O&M XML Schema, direct specialisations of OM_Observation are used in the INSPIRE application schemas, however the constraints introduced in the conceptual inheritance model should be observed (and could be validated using schematron or similar).

The feature catalogue for these feature types can be found in the O&M document, D2.9. However, for completeness each feature type is also discussed below, with an emphasis on usage in OF.

5.3.1.2.2. PointObservation (From Generic Conceptual Model)

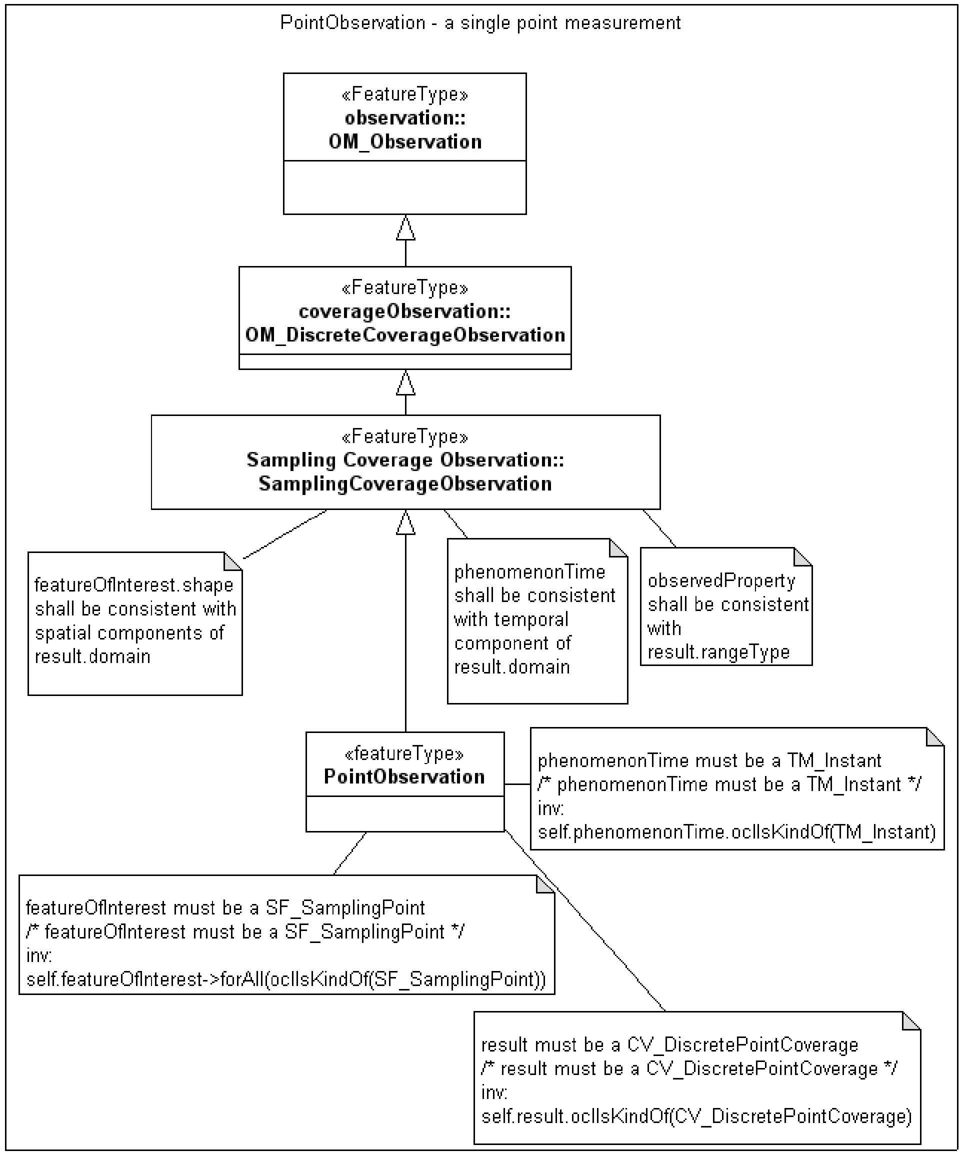


Figure 8 PointObservation

The PointObservation represents a single value measurement at a single point in time e.g. a manual one-off measurement of sea surface temperature.

PointObservation		
O&M Attribute/association	**Is constrained to be**	**Example: Single measurement of Sea Surface Temperature**
processUsed	Process (Section 5.3.1.2.10)	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingPoint	A SF_SamplingPoint at the geographic location of the measurement
phenomenonTime	TM_TimeInstant	A time instant (in ISO 8601 including time zone) e.g. 2012-01-30T10:30:00.00Z
observedProperty	ObservableProperty	The observed property should link to a vocabulary defining sea surface temperature,

		and should also indicate the units used in the result (e.g. Celsius).
Result	CV_DiscretePointCoverage	The result should be a single valued coverage recording an estimate of the observed property e.g. 22.2 (Celsius)
resultTime	TM_TimeInstant	The time the result was made available (e.g. published)

Table 2 Illustrated PointObservation

5.3.1.2.3. PointTimeSeriesObservation (From Generic Conceptual Model)

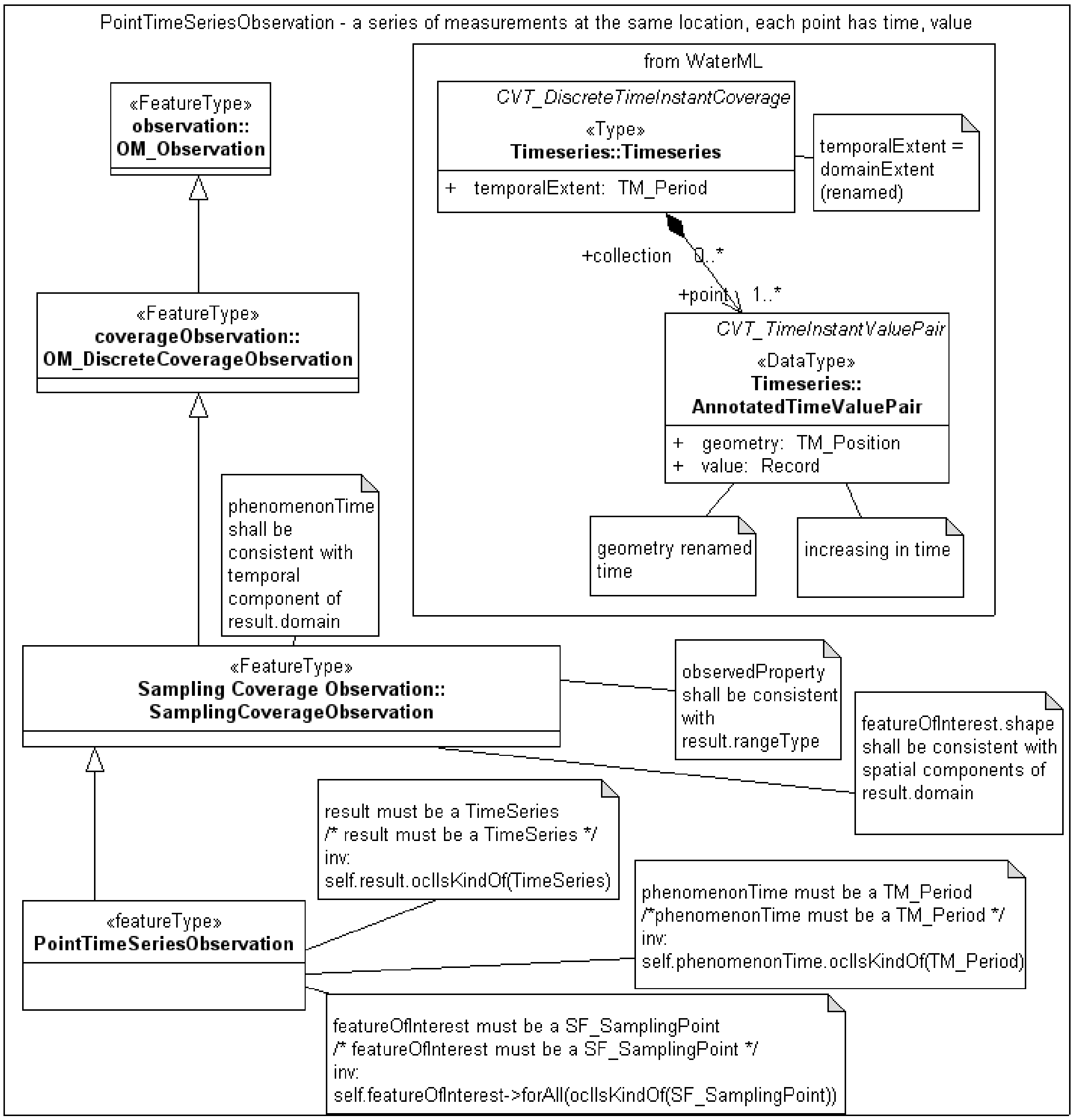


Figure 9 PointTimeSeriesObservation

The PointTimeSeriesObservation represents a series of measurements at the same point – a classic timeseries – e.g. regular measurements from a fixed station

PointTimeSeriesObservation		
		Example: Repeated measurements of Sea

O&M Attribute/association	Is constrained to be	Surface Temperature at the same location.
processUsed	Process	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingPoint	A SF_SamplingPoint at the geographic location of the measurement. It must be the same location for the entire time series. Note that in the case of fixed monitoring stations, the SF_SamplingPoint *could* be specialised in an extension schema to be a station feature type (or similar) to provide further information about the fixed station (e.g. a name). Although this is *not required*. Only the SF_SamplingPoint (with it's geometry) is required for OF. [See D2.9]
phenomenonTime	TM_TimePeriod	A time period (in ISO 8601) representing the start and end date/times of the time series.
observedProperty	ObservableProperty	The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).
result	TimeSeries	The result should be a set of time,value pairs encoded according to the Generic Conceptual Model (the WaterML XML encoding is used).
resultTime	TM_TimeInstant	The time the result was made available (e.g. published)

Table 3 Illustrated PointTimeSeriesObservation

Note that it may be convenient to have sampling features that are identifiable fixed stations rather than simple points. In this case, the mechanism to implement this would be to create a 'station' type (or similar) that is a specialism of SF_SamplingPoint. This station class could carry additional identification attributes and any other domain-specific information that is required.

5.3.1.2.4. MultiPointObservation (From Generic Conceptual Model)

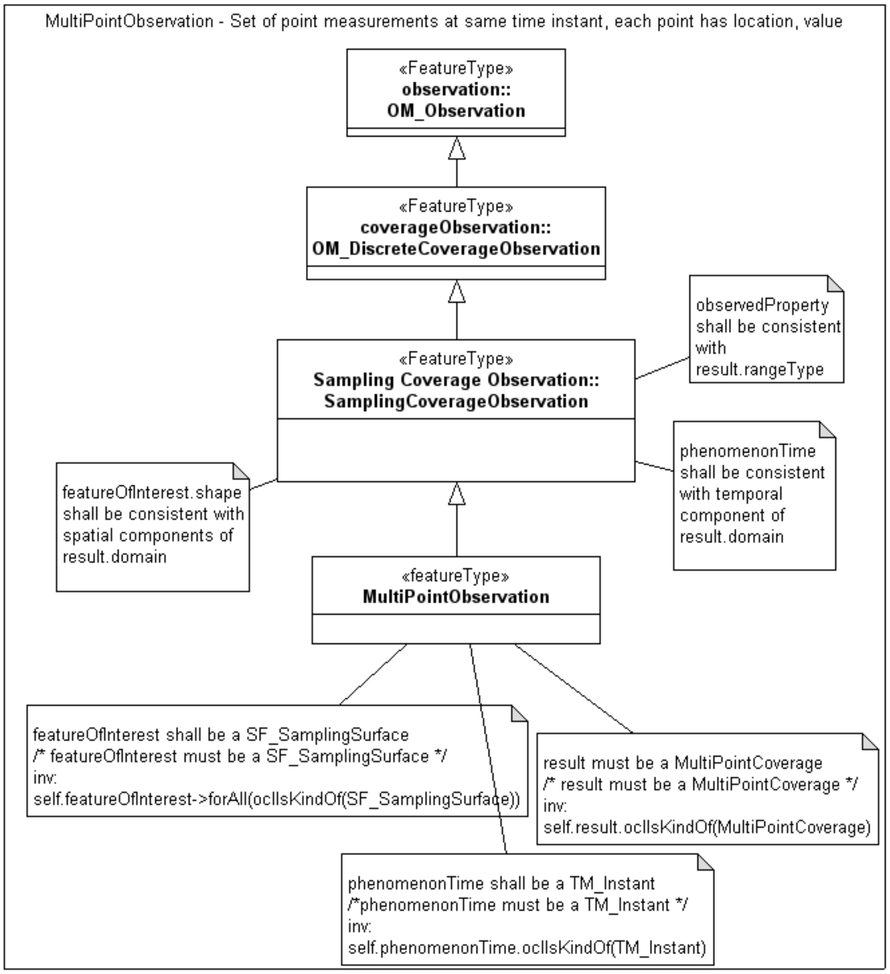


Figure 10 MultiPointObservation

The MultiPointObservation is a very specific type of Point-based observation. It is intended for the case where identical measurements are made at a set of discrete points *at the same time*. For example a sensor network reporting temperature at 10am. The points themselves are not on a grid but may be distributed in any manner – for example unevenly spaced around a coastline.

In this case the result of the observation is a GML MultiPointCoverage, which consists of a set of points (the domain) and a set of values (the rangeSet). (see GML 3.3.3).

MultiPointObservation		
O&M Attribute/association	**Is constrained to be**	**Example: Repeated measurements of Sea Surface Temperature at the same location.**
processUsed	Process	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingSurface or SF_SamplingSolid	A SF_SamplingSurface with a geometry that defines the *total extent* of the MultiPointObservation. (i.e. a bounding box or polygon that includes all the measurement locations).
phenomenonTime	TM_TimeInstant	A time instant (in ISO 8601) when the observations were taken (all measurements must be taken at the same time instant).
observedProperty	ObservableProperty	The observed property should link to a

		vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).
result	MultiPointCoverage	The result should be a GML MultiPointCoverage. For large result sets an out-of-band result (e.g. in binary) may be provided.
resultTime	TM_TimeInstant	The time the result was made available (e.g. published)

5.3.1.2.5. GridObservation (From Generic Conceptual Model)

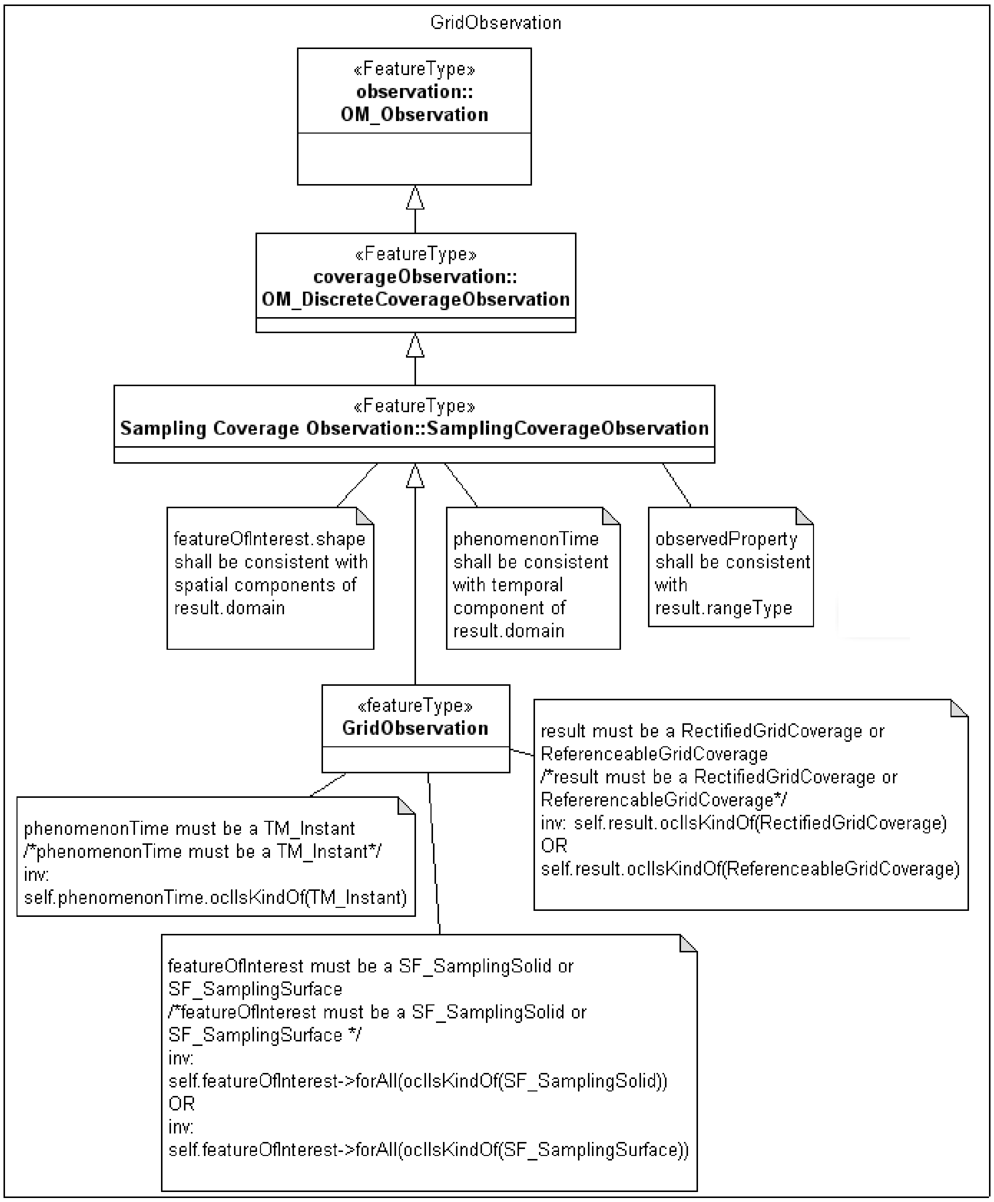


Figure 11 GridObservation

A GridObservation is a single grid of data – e.g. measurements taken by a satellite processed to be on a rectified geo-referenced grid (e.g. Level 3 processed data), or output from a numerical model.

The GridObservation is taken at a single snapshot in time. e.g. 10am, 30 January 2012.

GridObservation		
O&M Attribute/association	**Is constrained to be**	**Example: Grid of Ocean Colour**
processUsed	Process	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingSurface or SF_SamplingSolid (if there is a vertical dimension to the grid)	A SF_SamplingSurface that defines the *extent* of the Grid of data.
phenomenonTime	TM_TimeInstant	A time instant (in ISO 8601 including time zone) e.g. 2012-01-30T10:30:00.00Z which the Grid represents.
observedProperty	ObservableProperty	The observed property should link to a vocabulary defining Ocean Colour, and should also indicate the units used in the result (e.g the index type).
result	RectifiedGridCoverage or ReferenceableGridCoverage	The result should be a GML RectifiedGridCoverage or GML ReferenceableGridCoverage containing the grid points (as the domain of the coverage) and the observed ocean colour values (as the rangeSet of the coverage. For large grids an out-of-band result (e.g. in binary) may be provided.
resultTime	TM_TimeInstant	The time the result was made available (e.g. published)

Table 4 Illustrated GridObservation

5.3.1.2.6. GridSeriesObservation (From Generic Conceptual Model)

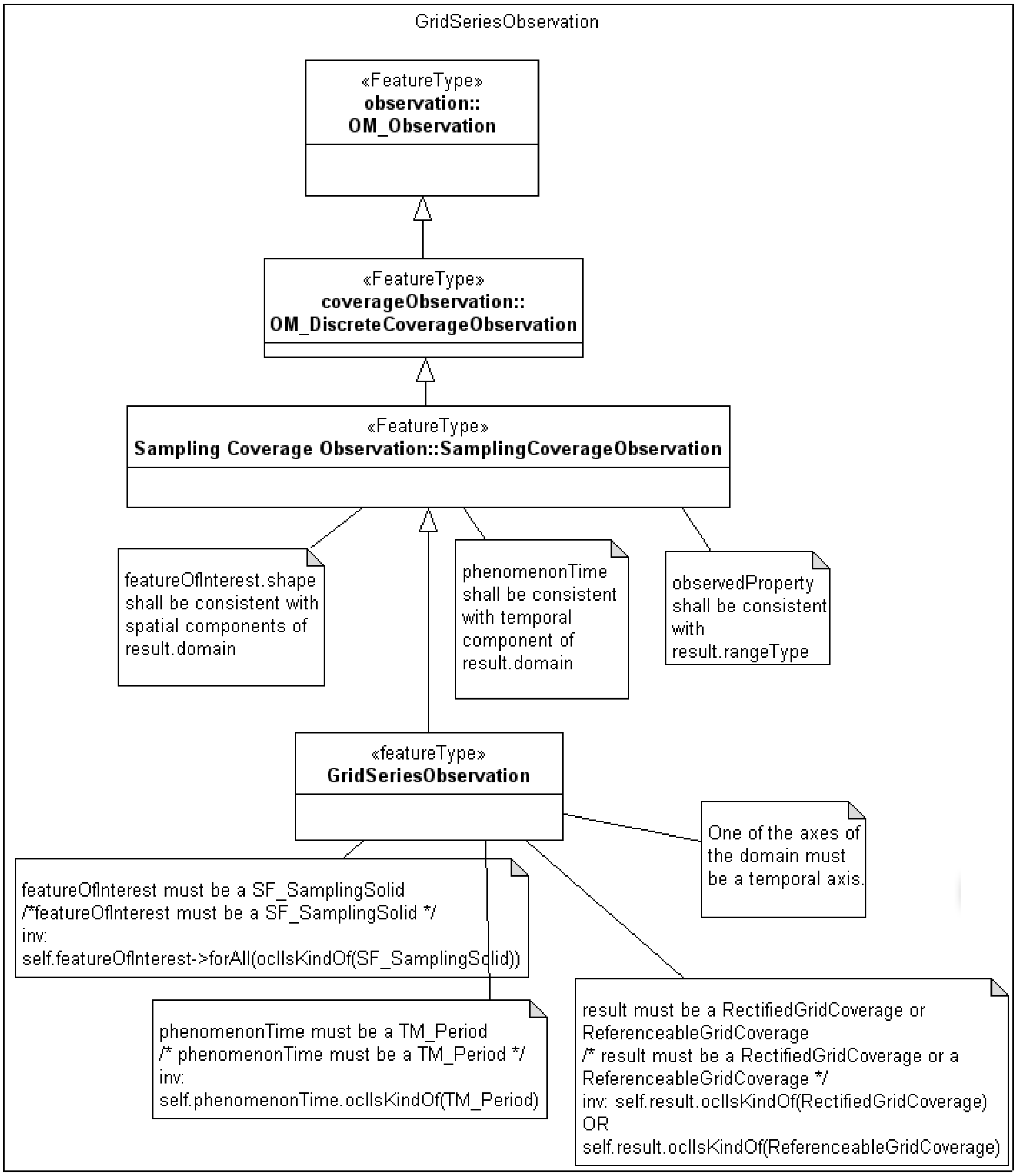


Figure 12 GridSeriesObservation

A GridSeriesObservation is similar to a GridObservation except it contains a series of grids for multiple, successive timesteps (e.g. a simulation/model run)

GridSeriesObservation		
O&M Attribute/association	**Is constrained to be**	**Example: Gridded model output showing predicted Sea Surface Temperature**
processUsed	Process	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingSurface or SF_SamplingSolid (if there is a vertical dimension to the grid)	A SF_SamplingSurface that defines the *extent* of the Grid of data.

phenomenonTime	TM_TimePeriod	A time period (in ISO 8601) representing the start and end date/times of the model run.
observedProperty	ObservableProperty	The observed property should link to a vocabulary defining Sea Surface Temperature, and should also indicate the units used in the result.
result	RectifiedGridCoverage or ReferenceableGridCoverage	The result should be a GML RectifiedGridCoverage or GML ReferenceableGridCoverage containing the grid points (as the spatio-temporal domain of the coverage) and the observed sea surface temperature values (as the rangeSet of the coverage. Note that one of the axes of the grid coverage domain must be a temporal axis as GridSeriesObservation is a type of time series. For detailed encoding of GML coverage types see GML 3.3.3. For large grids an out-of-band result (e.g. in binary) may be provided.
resultTime	TM_TimeInstant	The time the result was made available (e.g. published)

Table 5 Illustrated GridSeriesObservation

5.3.1.2.7. ProfileObservation (Informative in OF, from Generic Conceptual Model)

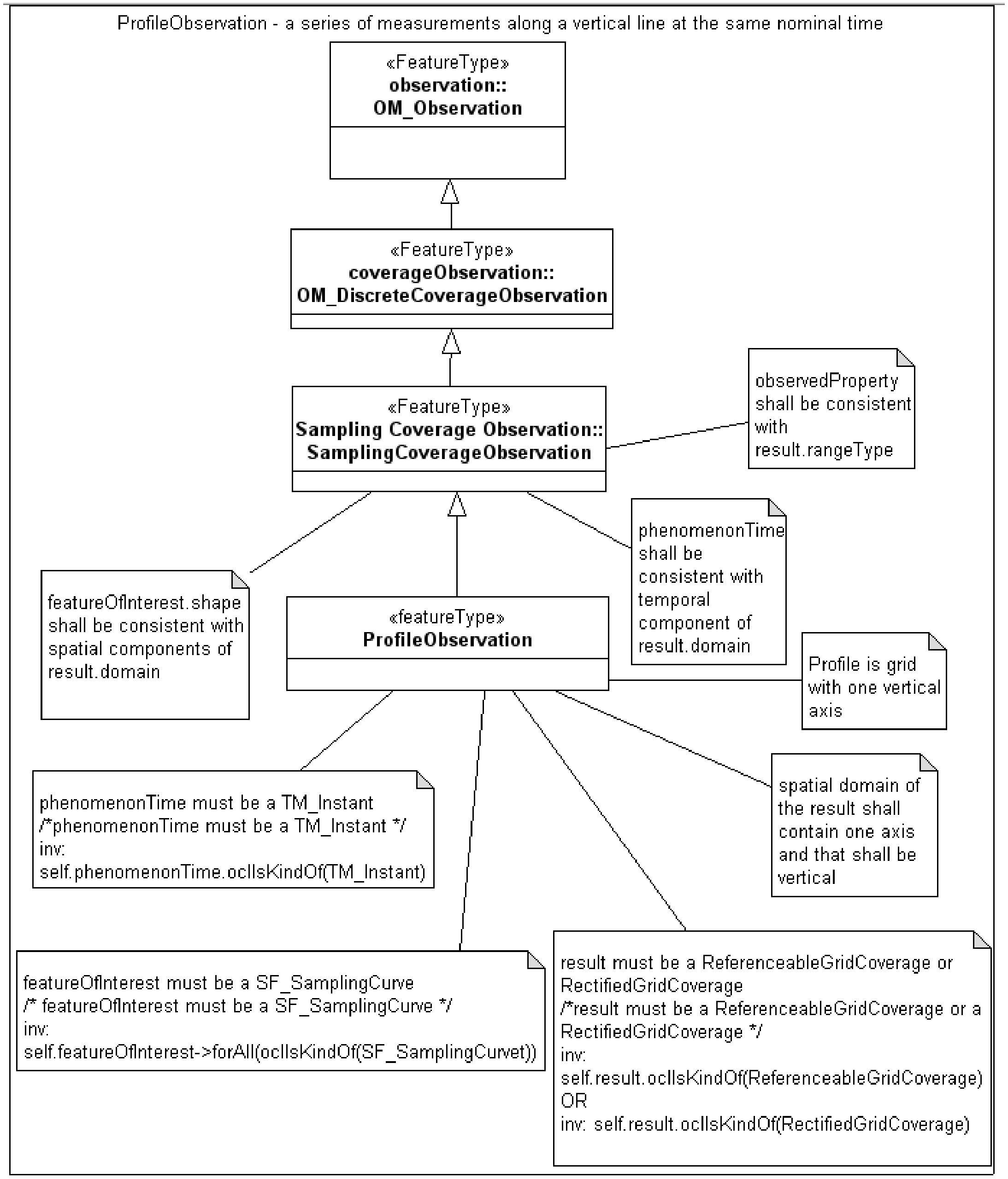


Figure 13 ProfileObservation

The ProfileObservation is informative only. A ProfileObservation represents a set of points along a vertical axis with a measurement value at each point on the profile. The measurements are all nominally made at the same time for the entire profile. The profile is encoded as a one dimensional grid coverage, again using the GML coverage models.

ProfileObservation		
O&M Attribute/association	**Is constrained to be**	**Example: Salinty depth profile.**
processUsed	Process	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingCurve	A SF_SamplingCurve with a geometry that defines the geometry of the profile.

phenomenonTime	TM_TimeInstant	A time instant (in ISO 8601) when the observations were taken (all measurements must be taken at the same time instant).
observedProperty	ObservableProperty	The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).
result	RectifiedGridCoverage or ReferenceableGridCoverage	The result should be a GML RectifiedGridCoverage or ReferenceableGridCoverage with a single spatial dimension (which should be vertical). For large result sets an out-of-band result (e.g. in binary) may be provided.
resultTime	TM_TimeInstant	The time the result was made available (e.g. published)

Table 6 Illustrated ProfileObservation

5.3.1.2.8. TrajectoryObservation(Informative in OF, from Generic Conceptual Model)

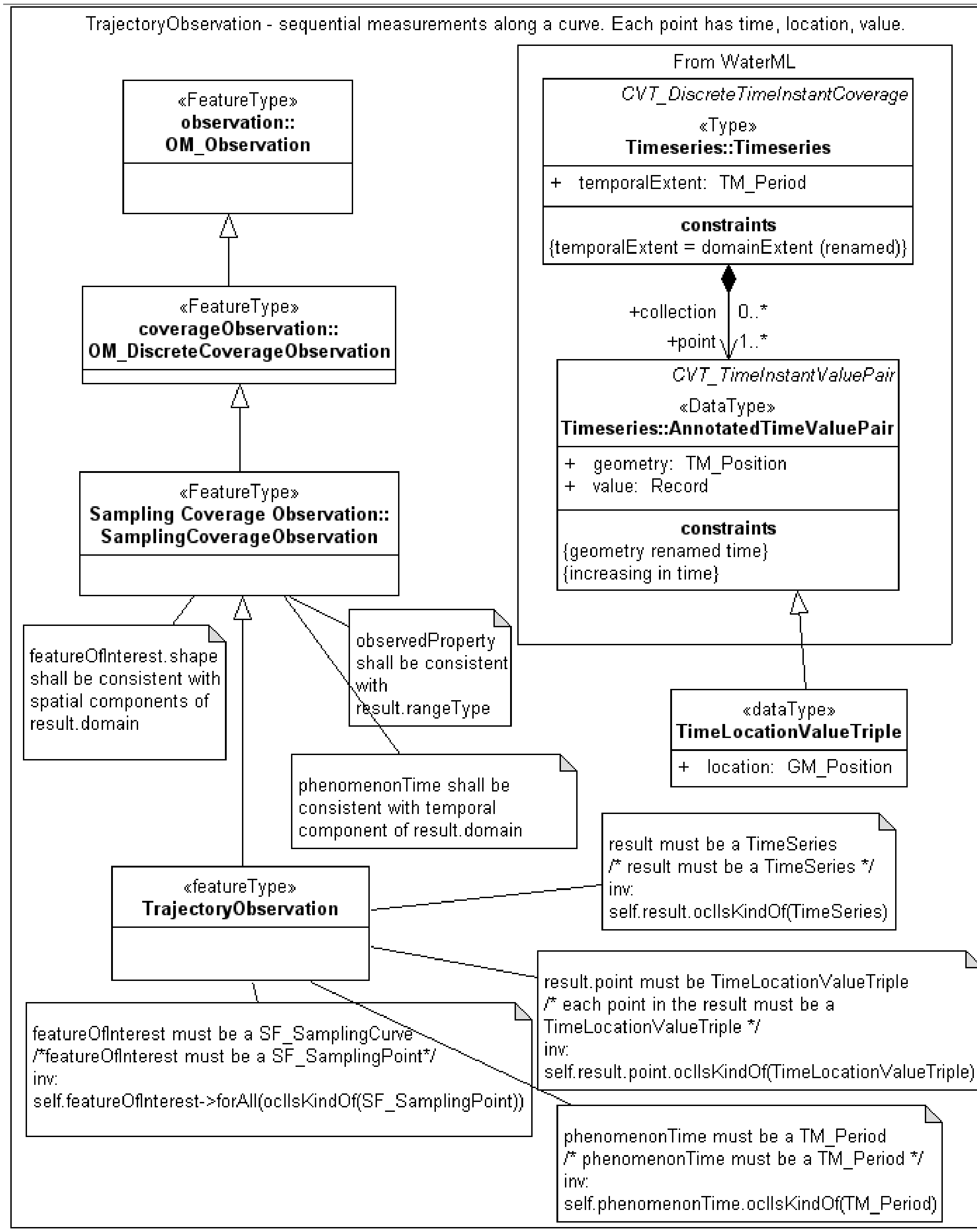


Figure 14 TrajectoryObservation

A TrajectoryObservation represents a series of measurements along a trajectory. For example along a ship's track. Each measurement is made at a separate point along the trajectory and at a separate time. The result is therefore a set of time, location, value triples.

TrajectoryObservation		
O&M Attribute/association	**Is constrained to be**	**Example: Sea Surface Temperature along a ship's track.**
processUsed	Process	Process instance links to information about the responsible party, documented process etc.
featureOfInterest	SF_SamplingCurve	A SF_SamplingCurve with a geometry that defines the geometry of the trajectory
phenomenonTime	TM_TimePeriod	A time period (in ISO 8601) representing the start and end date/times of the trajectory.
observedProperty	ObservableProperty	The observed property should link to a vocabulary defining sea surface temperature, and should also indicate the units used in the result (e.g. Celsius).
result	TimeSeries, with triple values	The result should be a set of Location, Time, Value triples encoded according to the conceptual model and application schema (an extension of the WaterML time,value pair encoding is used to model time,location, value triples).
resultTime	TM_TimeInstant	The time the result was made available (e.g. published).

Table 7 Illustrated Trajectory Observation

5.3.1.2.9. ObservationSet, PointObservationCollection

The GenericConceptual model contains an *ObservationSet* type which is simply a set of *OM_Observations* (or specialisations thereof).

A specialised type of collection *PointObservationCollection* is also present in this model. This is to satisfy the common use case where a set of otherwise independent *PointObservations* should be logically grouped together. An example of this in OF would be a set of *PointObservations* taken by the same vessel on a cruise around a coastline.

Recommendation 7 The PointObservationCollection shall be used in OF when a set of PointObservation features form a coherent set

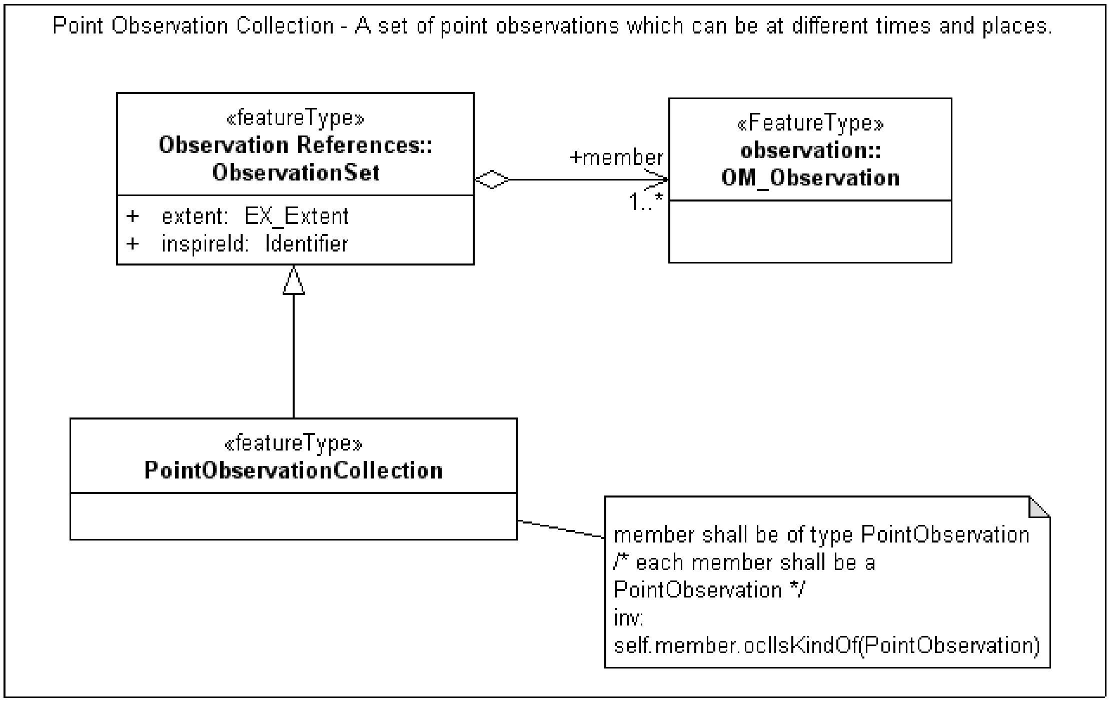


Figure 15 PointObservationCollection

5.3.1.2.10. Process

Observations and Measurements provides an abstract class '*OM_Process*' which describes the process used to acquire a measurement value. INSPIRE specialises this into *Process* which provides a structured way to describe the process, in particular to point to external documentation (online or offline) that describes the process and to refer to parameters used in the process.

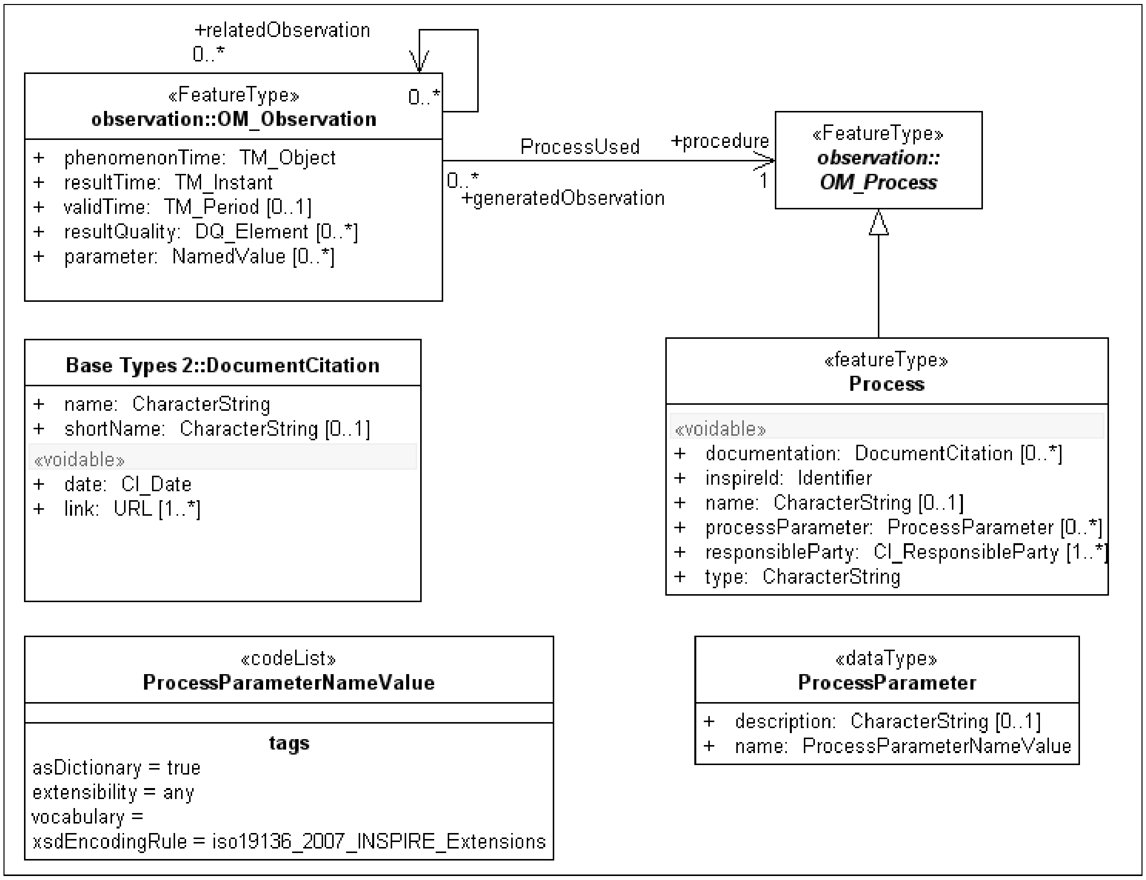


Figure 16 INSPIRE specialisation of OM_Process

The *ProcessParameter* type may be used to extend *Process* descriptions where it is necessary to capture event-specific parameters or settings that need recording each time the *Process* is used. These parameters shall be identified using HTTP URIs.

This is really a generic extension mechanism to allow key aspects of any type of process to be exposed while still using the basic, generic INSPIRE *Process* type.

For example, say it's vitally important to know the ambient air temperature every time a particular type of observation is made with a certain instrument. You could add ‘ambient temperature' as a parameter in the generic Process description, and then when you have a particular Observation you add in an om:parameter saying the ambient temperature was 20 degrees celsius when this particular observation was made.

e.g.
Process.processParameter = ProcessParameter(name=*http://some.authority/ambientTemp*, description=”the ambient air temperature around the instrument”)

The Observation itself shall indicate what the ambient temperature was for a particular observation via the om:parameter NamedValue mechanism:

OM_Observation.om:parameter = NamedValue(name=”http://some.authority/ambientTemp”, value=22.3)

This example is purely illustrative, the use of *ProcessParameters* is entirely discretionary but the mechanism is made available so that there is a way to capture essential process information in the generic INSPIRE *Process* type without having to extend the data model for each process.

Recommendation 8	Parameters of a process that are fundamental to understanding the observation but are not explicitly defined may be described using the ProcessParameter class

TG Requirement 2	For every ProcessParameter a process has, a corresponding om:parameter shall be included in the observation. The om:parameter.name shall mirror the name of the ProcessParameter.

5.3.1.2.11. References to Environmental Monitoring Features.

Environmental monitoring features (facilities, networks etc) are described by the Environmental Monitoring Facilities (EF) specification. However it is often important to be able to identify the environmental monitoring facility used in a particular observation. To ensure consistent referencing between Observations and Environmental Monitoring Features there is a procedure described in [DS-D2.9].

In the case where the Environmental Monitoring Facility used is co-incident with the spatial sampling feature then efforts shall be made to ensure that the coordinate geometries of these spatial object types are consistent with each other.

Where differences in location/geometry do occur (either actual differences, or differences of accuracy), the spatial sampling feature geometry described in the Observation shall take precedence and can be assumed to be the correct sampling location of the Observation.

5.3.1.2.12. Observed properties and OF Vocabularies.

This INSPIRE OF data specification does not specify which phenomena of the ocean (e.g. temperature, salinity) are being observed. The phenomena are described by the 'observedProperty' attribute of the Observation.

In addition the OF theme does not specify what phenomena should be fall within scope of the theme, it simply states that where there is a mandate to measure or observe a parameter in accordance with European Legislation is should be exchanged according to this specification.

However it is essential to enable unambiguous identification of the property that is being observed (e.g. temperature, salinity). Definitions of such terms are subject to strict governance by external organisations and these definitions are recognised by INSPIRE.

Two external vocabularies are recognised as suitable for identifying the observed property of an OF observation. These are:

- BODC (British Oceanographic Data Centre) P01 Parameter Usage vocabulary
- CF (Climate and Forecast) Standard Names

Both vocabularies are in widespread used throughout the EU and are subject to strict governance procedures.

IR Requirement
Annex IV, Section 14.3
Theme-specific Requirements

The observed property of an OM_Observation shall be identified by an identifier from the BODC P01 Parameter Usage or Climate and Forecast Standard Names vocabularies.

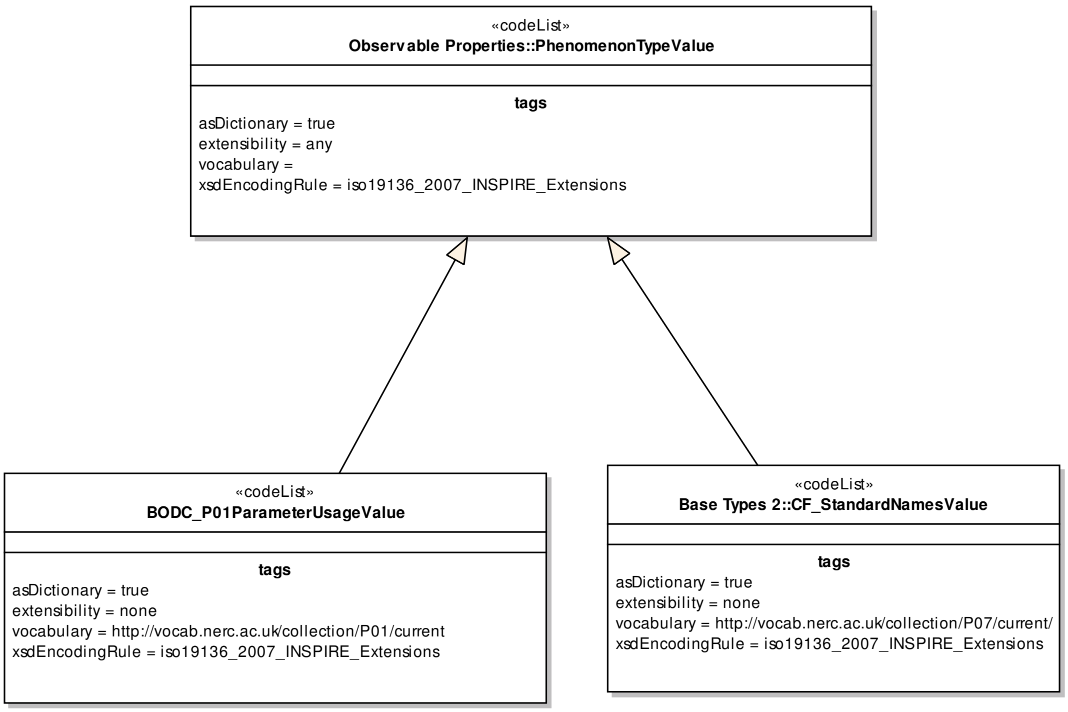


Figure 17 Codelists prescribed in OF

In addition, the GCM provides a framework for adding additional information to the observable property, for example, a statistical measure e.g. 'Daily Mean' temperature, or a value-based constraint e.g. radiance within a particular wavelength band. Further detailed discussion of this model is given in D2.9.

Recommendation 9 The ObservableProperty model in D2.9 should be used to provide qualified descriptions of the observed property with reference to the external vocabularies mandated in this document.

5.3.1.3. Consistency between spatial data sets

The O&M SamplingCoverageObservation consistency constraints are used to ensure that:
- the rangeType of the observation result is consistent with the phenomenon measured
- the phenomenon time of the observation is consistent with the temporal aspect of the coverage result
- the spatial components of the sampling feature shall be consistent with the observation result.

In addition, the Specialised Observation consistency constraints are used to ensure that:
- the feature of interest of the observation is an appropriate spatial sampling feature
- the phenomenonTime of the observation is either a time instant or period, as appropriate
- the result of the observation is the appropriate coverage type

For the individual classes, the consistency rules for the Specialised Observations in the Generic Conceptual Model are as follows:

PointObservation

- The feature of interest must be a sampling point, SF_SamplingPoint
- The phenomenon time must be a single time instant, TM_Instant
- The result must be a CV_DiscretePointCoverage –a MultiPointCoverage with a single point in the domain.
- This is consistent with a *single* measurements at a single point in time and space

PointTimeSeriesObservation

- The feature of interest must be a sampling point, SF_SamplingPoint
- The phenomenon time must be a time period, corresponding to the start and end times of the observation event, TM_Period
- The result must be a TimeSeries (from WaterML).
- This is consistent with a *time series* of measurements at a single point in space

MultiPointObservation

- The feature of interest must be an area, SF_SamplingSurface
- The phenomenon time must be a single time instant, TM_Instant - values at all points are measured at the same time.
- The result must be a MultiPointCoverage.
- This is consistent with *multiple* observations at the *same time*.

GridObservation

- The feature of interest must be a surface or solid, SF_SamplingSurface or SF_SamplingSolid
- The phenomenon time must be a single time instant, TM_Instant
- The result must be a RectifiedGridCoverage or ReferenceableGridCoverage.
- This is consistent with a *single* grid of data at an instant in time.

GridSeriesObservation

- The feature of interest must be a surface or solid, SF_SamplingSurface or SF_SamplingSolid
- The phenomenon time must be a time period, corresponding to the start and end times of the observation event, TM_Period.
- The result must be a RectifiedGridCoverage, or ReferenceableGridCoverage, with multiple timesteps in the domain.
- This is consistent with *multiple* timesteps of data on the same spatial grid.

ProfileObservation

- The feature of interest must be a line, SF_SamplingCurve
- The phenomenon time must be a single time instant, TM_Instant
- The result must be a RectifiedGridCoverage, or ReferenceableGridCoverage, with a single point in the temporal domain and one vertical dimension.
- This is consistent with a *vertical profile* of data at an instant in time.

TrajectoryObservation

- The feature of interest must be a curve, a SF_SamplingCurve
- The phenomenon time must be a time period, TM_Period.
- The result must be a TimeSeries where each point in the TimeSeries is a TimeLocationValueTriple..
- This is consistent with measurements following a *trajectory* along a ship's track.

5.3.1.4. Modelling of object references

The phenomena types (e.g. Temperature, Salinity) referenced by the ‘observedProperty’ attribute of the Specialised Observations are governed in external vocabularies as described in Section 5.3.1.2.12 and described according to the Observable Property model in the GCM.

5.3.1.5. Geometry representation

Art. 12(1) of Regulation 1089/2010 restricts the value domain of spatial properties to the Simple Feature spatial schema as defined in the *OpenGIS® Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture, version 1.2.1*, unless specified otherwise for a specific spatial data theme or type.

The geometry of an OF Observation is a key attribute (shape) of the sampling feature. The sampling feature is the feature of interest (such as an SF_SamplingSurface).

Spatial elements are also an intrinsic part of the *result* of OF Observations – i.e. the 'domain' of the coverage – e.g. the point or points for which measurements exist. So for example, a PointTimeSeriesObservation represents a timeseries of data at a specific spatial location, and a GridObservation represents measurements at a number of discrete spatial locations. The geometry of the grid in is not necessarily simple.

5.3.1.6. Temporality representation

There is an important temporal aspect to the Specialised Observations used in OF. The 'phenomenonTime' of the Observation represents the bounding time envelope or instant for any observation event. For time series features (PointTimeSeriesObservation, GridSeriesObservation, TrajectoryObservation) the temporal element of the domain of the coverage result will describe the full temporal detail of the observation (exactly when each measurement was made).

5.3.2 Feature catalogue

Feature catalogue metadata

Application Schema	INSPIRE Application Schema Oceanographic Geographical Features
Version number	3.0

Types defined in the feature catalogue

Type	Package	Stereotypes
BODC_P01ParameterUsageValue	Oceanographic Geographical Features	«codeList»

Code lists

5.3.2.1.1. BODC_P01ParameterUsageValue

BODC_P01ParameterUsageValue	
Name:	BODC P01 Parameter Usage
Definition:	Definitions of phenomena observed in oceanography.
Extensibility:	open
Identifier:	http://vocab.nerc.ac.uk/collection/P01/current
Values:	The allowed values for this code list comprise the values specified in "British Oceanographic Data Centre (BODC) Parameter Usage Vocabulary" and additional values at any level defined by data providers.

INSPIRE governed code lists are given in Annex C.

5.3.3 Externally governed code lists

The externally governed code lists included in this application schema are specified in the tables in this section.

5.3.3.1. Governance and authoritative source

Code list	**Governance**	**Authoritative Source (incl. version[2] and relevant subset, where applicable)**
CF_StandardNamesValue	CF Governance Committee and CF Standard Names Committe (representatives from multiple data centres)	British Oceanographic Data Centre
BODC_P01ParameterUsageValue	British Oceanographic Data Centre in association with SeaVox vocabulary governance (a cross-organisational governance mailing list)	British Oceanographic Data Centre

5.3.3.2. Availability

Code list	**Availability**	**Format**
CF_StandardNamesValue	*http://vocab.nerc.ac.uk/collection/P07/current/* *http://cf-pcmdi.llnl.gov/documents/cf-standard-names*	SKOS/RDF, XML, HTML
BODC_P01ParameterUsageValue	*http://vocab.nerc.ac.uk/list/P01/current/* *http://vocab.nerc.ac.uk/collection/P01/current*	SKOS

5.3.3.3. Rules for code list values

Code list	**Identifiers**	**Examples**
CF_StandardNamesValue	n/a	*http://vocab.nerc.ac.uk/collection/P07/current/CFSN0413*
BODC_P01ParameterUsageValue	Add unique code to *http://vocab.nerc.ac.uk/collection/P01/current/* Unique codes can be found in rdf:about tags in	http://vocab.nerc.ac.uk/collection/P01/current/ASLTZZ01

[2] If no version or publication date are specified, the “latest available version” shall be used.

	SKOS version of the list.	

Code list	**Labels**	**Examples**
CF_StandardNamesValue	The string contained in SKOS preflabel e.g <skos:prefLabel>relative_humidity </skos:prefLabel>	relative_humidity used for relative humidity
BODC_P01ParameterUsageValue	The string contained in SKOS prefLabel e.g. from: <skos:prefLabel xml:lang="en">Absolute salinity of the water body</skos:prefLabe The label is: “Absolute salinity of the water body”	absolute salinity of the water body