Sensor Data and Metadata Standards Review for UKCEH

Author(s)	Stuart Williams (Epimorphics Ltd.) Jan Rosecký (Epimorphics Ltd.)	December 2019
Reviewer(s)	Matt Fry (UKCEH) Oliver Swain (UKCEH) Simon Stanley (UKCEH) Mike Brown (UKCEH)	

1 Executive Summary

This report has been prepared by Epimorphic Ltd. as part of the ENTRAIN project[1] (NERC grant number NE/S016244/1) which is a feasibility project within the "NERC Constructing a Digital Environment Strategic Priorities Fund Programme". The UK Centre for Ecology and Hydrology (UKCEH) operates a number of sensor networks which together build a corpus of environmental data curated and maintained by UKCEH. This data supports scientific research and policy making.

There is a need to make the data gathered by sensor networks available to the scientific community, both within UKCEH and more widely, in ways that are easily accessible and where the data can be easily understood. The work undertaken in generating this report reviews and summarises a number of technical specifications originating with the ISO, OGC and W3C standards bodies in the field of earth observations and sensor networks. It summarises and discusses the core concepts embodied in these specifications; relates them to data sets maintained by UKCEH particularly data arising from the COSMOS-UK, NRFA and Thames Initiative networks; provides illustrative example that show how the specifications could be applied to the exchange of both sensor data and associated metadata covering observed features and properties, observation processes and procedure, sensor types, instances and deployments.

The central concept of the specifications reviewed is that of an act of Observation, which estimates the value of some property of some a feature-of-interest (an abstraction of some real-world phenomenon). The observation acts as a hub for the expression of *what* is being observed (some property of some feature); *how* the observation was made (procedure used) and the sensors or facilities used; *when* the observation was made (result time and phenomenon time); and the observation result itself. However, it is generally the case that the raw specifications provide a lot of flexibility in the way that they can be used or applied. This leads to some diversity in the way that they are used and a need for guidance and profiles around which common practices can emerge. In particular, in practice it is evident that aspects of procedure, result units, substances of interest (soil, water) and constraints (measurement height/depth) can be entailed by a combination of feature-of-interest and observed property. This leads to the Complex Property Model of Leadbetter and Vodden which makes for semantically grounded property definitions which aid in their discovery for reuse and in their interpretation.

This review is a precursor to the development of a JSON based transfer format for time series and associated metadata. The intention is that this format will be simple to read and write and that it will be conceptually grounded in some of the specifications reviewed here. In particular we recommend the development of a JSON-LD based format built around core concepts drawn jointly from ISO19156 Observations and Measurements and W3C Semantic Sensor Networks (SSN)/Sensor, Observation, Sample and Actuator (SOSA); the use of the Complex Property Model for semantically grounded property descriptions; OGC Sensor ML for the description of sensor instances and types; and INSPIRE Environmental Monitoring Facilities for describing sites and their monitoring capabilities. For time series representation we recommend the use of observation collections (a SSN/SOSA extension) which also serve as a point of attachment for property/values shared by all observations in a collection.

[1] http://www.ceh.ac.uk/our-science/projects/entrain

Table Of Contents

2 Introduction

The UK Centre for Ecology and Hydrology (UKCEH[2]) “is a world-class research organisation focusing on land and freshwater ecosystems and their interaction with the atmosphere.”

> *“Uniquely, CEH integrates UK-wide observation systems and curiosity driven research, from the smallest scale of genetic diversity to large-scale, whole-Earth systems. We work across disciplines and facilitate academic, public, private and voluntary sector partnerships. CEH’s extensive, long-term monitoring, analysis and modelling deliver UK and global environmental data, providing early warnings of change and management solutions for our land and freshwaters.”*

UKCEH therefore has a remit to monitor the environment through the collection of observational data using sensor networks such as a COMOS[3]. Such monitoring leads to an accumulation of observational data which needs to be shared within UKCEH and potentially with other agencies and organisations. The effective sharing of data requires common data standards (small ‘s’) and established communities of practice around the use of those standards to support the correct interpretation of data.

The correct interpretation of recorded data will require much more than the readings that were recorded by the sensor. There is a need for contextual information about what was being measured and the units in which it is being recorded; temporal information about the period over which an aggregated result is generated and the nature of the aggregation; indication of data quality assessment to accompany readings; information about the observation process and any sampling that might have occured; information about the geographic feature or phenomenon being observed - which may be a stretch of a river or air quality for a particular city or town; information about the sensor(s) used in making a measurement and in some cases information about the deployment, calibration and maintenance history of the sensor. Figure 1 below offers a grouping of these topic areas as a way to think about the coverage of related specifications and in particular the specifications reviewed in section 4.

This report is part of a programme of work intended to develop a common format and associated practices for the publications and sharing of UKCEH environmental observation and monitoring data along with sufficient metadata for accurate interpretation as minimum. In addition there is a potential need to be able to include or reference other relevant contextual data: process details; sensor lifecycle events (deployment, calibration, maintenance…); observed features (river segments, monitoring sites, sampling point, retained samples) their locations and inter-relationships (segments and nodes in a river network, upstream hydrological catchments…). As a first step, this report provides an overview of several standards and specifications targeted on the recording and representation of observational and time series data (section 4); it identifies and discusses common themes and issues that are approached differently in these specifications (section 5); it moves on to illustrate their application to three existing UKCEH data sets (sections 6 and 7). Finally it seeks to set a direction for the next phase of work (sections 8 and 9) to develop a concrete transfer format specification to address the requirements discussed more fully in the next section.

---

[2] https://www.ceh.ac.uk/
[3] https://cosmos.ceh.ac.uk/

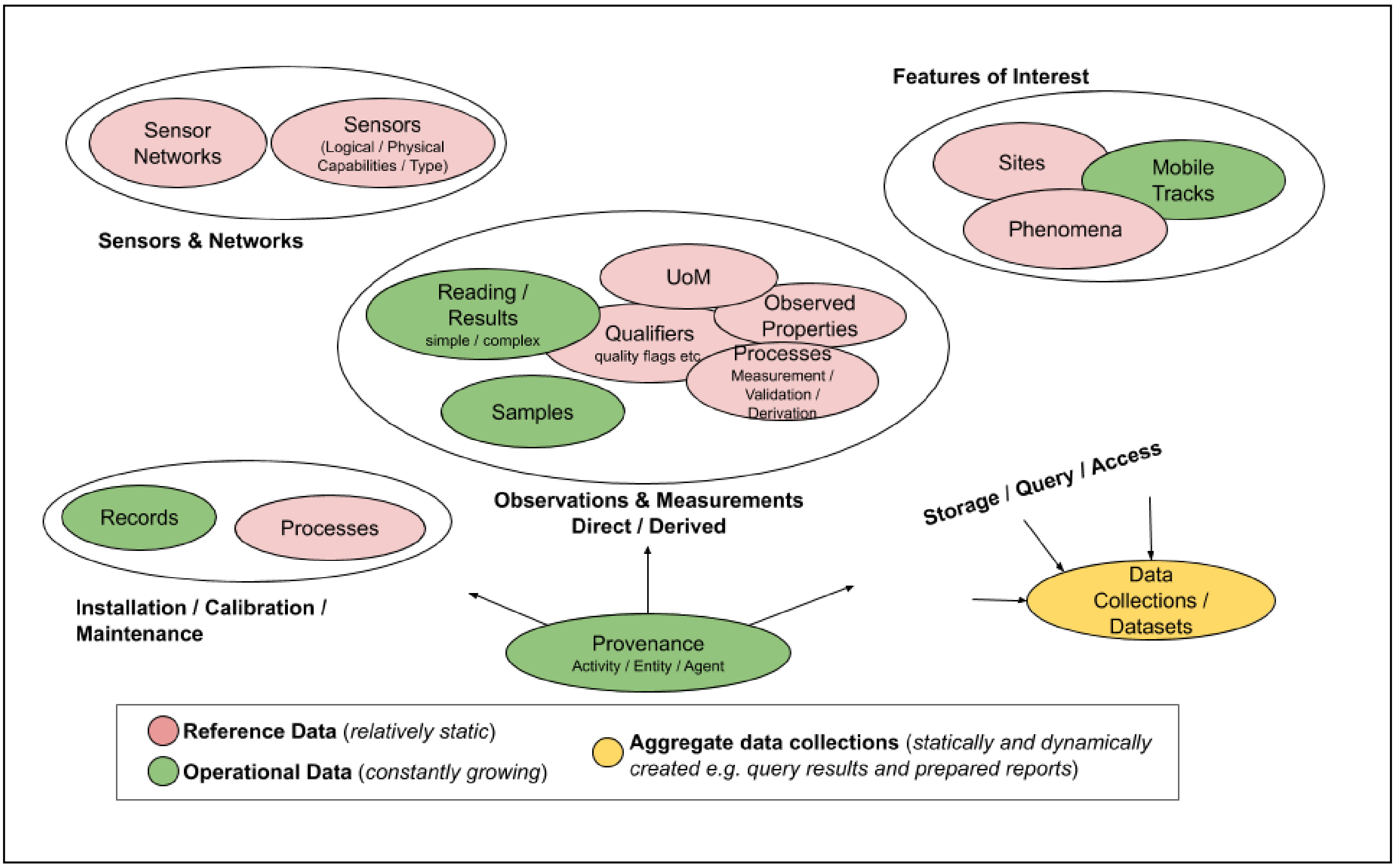


Figure 1 Specification Topic Areas

3 Requirements

3.1 Framing

UKCEH operate a number of environmental monitoring sensor networks that collect observation data from multiple sites or stations. Typically there are multiple sensors installed at a site that monitor a number of phenomena such as soil and air temperature, soil-moisture content, air speed and direction, river flow rates and depths and many more.

Data from these networks is collected on a regular basis and organised into time series that can be used to show how the state of the observed phenomena changes over time at each of the places where it is observed. In order to interpret data collected by these networks, there is a need to be able to provide accompanying metadata about the sensors and methods used in data collection, information about the features and phenomena observed and the sites where they are located.

UKCEH are seeking to develop a common approach to the publishing and sharing of its sensor network data in ways that also carry or make reference to accessible metadata to support themselves and third parties in the discovery and interpretation of sensor data, leading to scientific conclusions and potentially environmental policy and decision making. Interoperable data access and sharing in this way requires a number of elements:

- A common data model for the expression of data and metadata. At the data level this will need to cover basic information about what is being observed/monitored and the results of such observations; at the metadata level there is a need for information about individual sensors, their types and capabilities, the sites where they are deployed and the features/phenomena monitored at those sites.
- Guidance on the concrete application of the data model in the context of the data sets that UKCEH curate.
- Across earth and environmental sciences domain there are collections of reference data (features, phenomena, materials, regions…) where it makes better sense to build community and share the effort of curating and maintaining referenced data. Guidance should extend to concrete choices about sources of key reference data, and practices for publishing UKCEH reference data for use by others.
- A concrete transfer syntax for the exchange of sensor data and metadata which arises from a common data model (1st bullet above). Ideally the transfer syntax should be capable of expressing both bulk data, such as say an annual snapshot of the COSMOS data, and more dynamic responses to API (application programming interface) requests made in support of interactive applications.
- API access patterns to support query and access of the data held. This includes access to metadata for the purposes of data discovery - what data is available about phosphate levels in UK rivers - as well as access to the monitoring data itself.

There is an overarching requirement to maintain simplicity and to be able to provide RESTful web interfaces that provide responses in an easily understood, idiomatic JSON based format.

This document reviews and summarizes a number of relevant data and metadata specifications in the context of three UKCEH sensor networks and their associated data sets. Section 4 gives an overview of several relevant specifications. Generally it is the case that the raw specifications provide a lot of flexibility in the way that they can be used or applied. However, this leads to some diversity in the way that they are used and a need for guidance and profiles around which common practices can emerge; Section 5 draws out a number of cross cutting themes and issues that are addressed in different ways by different specifications; Section 6 provides an outline of the three UKCEH sensor networks and data sets used to focus this investigation; Section 7 develops some examples of how the specification reviewed in section 4 could be applied to the UKCEH networks and data introduced in section 6. Section 8 draws together our high-level recommendations and begins to sketch a direction for a JSON based exchange format to be developed further in the next stage of the work. Finally section 9 outlines proposed next steps,

3.2 Requirements

UKCEH has the following requirements for a transfer syntax for observational time series data and accompanying metadata:

- Data centric meaning that the format is centred on the exchange of time series data and enough surrounding information for it to be interpreted safely - what properties of what features are being measured; when and where the measurements took place. Additional information about individual sensors making measurements (and their type) or the computational processes by which data is qualified, cleaned or aggregate are of interest, but are not central.

- Simplicity - first and foremost the format needs to be consumable by a wider range of users not versed in complex data models. The format should be human readable and an intuitive reading of data expressed in the format should generally be accurate to people familiar with the environmental domain to which it relates.

- Primarily time series captured by sensors that are stationary whilst in use. Support for mobile sensors would be useful, but not if it generates significant complexity.

- Support for JSON/REST API style by providing and using an idiomatic JSON based format.

- Alignment with the practices adopted by other groups (e.g. NERC family, CSIRO and others) and with existing standards and profiles - whilst maintaining an overall simplicity.

- Discovery metadata - data sets need to be discoverable. It needs to be possible to distill data and metadata from a collection of time series (could be just one) into metadata structures that enable data discovery. For example, thematic coverage, spatial and temporal extent (aka envelope), feature and property coverage, monitored substances and/or species….

4 Standards Overview

Figure 2 provides a high-level overview of the major concepts covered by the standards and specifications reviewed in this section. As presented here, this diagram presents something of a synthesis derived from the specifications reviewed in the following subsections. At the head of each subsection we will enumerate the central concepts for the specification covered by that subsection. The remainder of this section provides a quick tour of conceptual space.

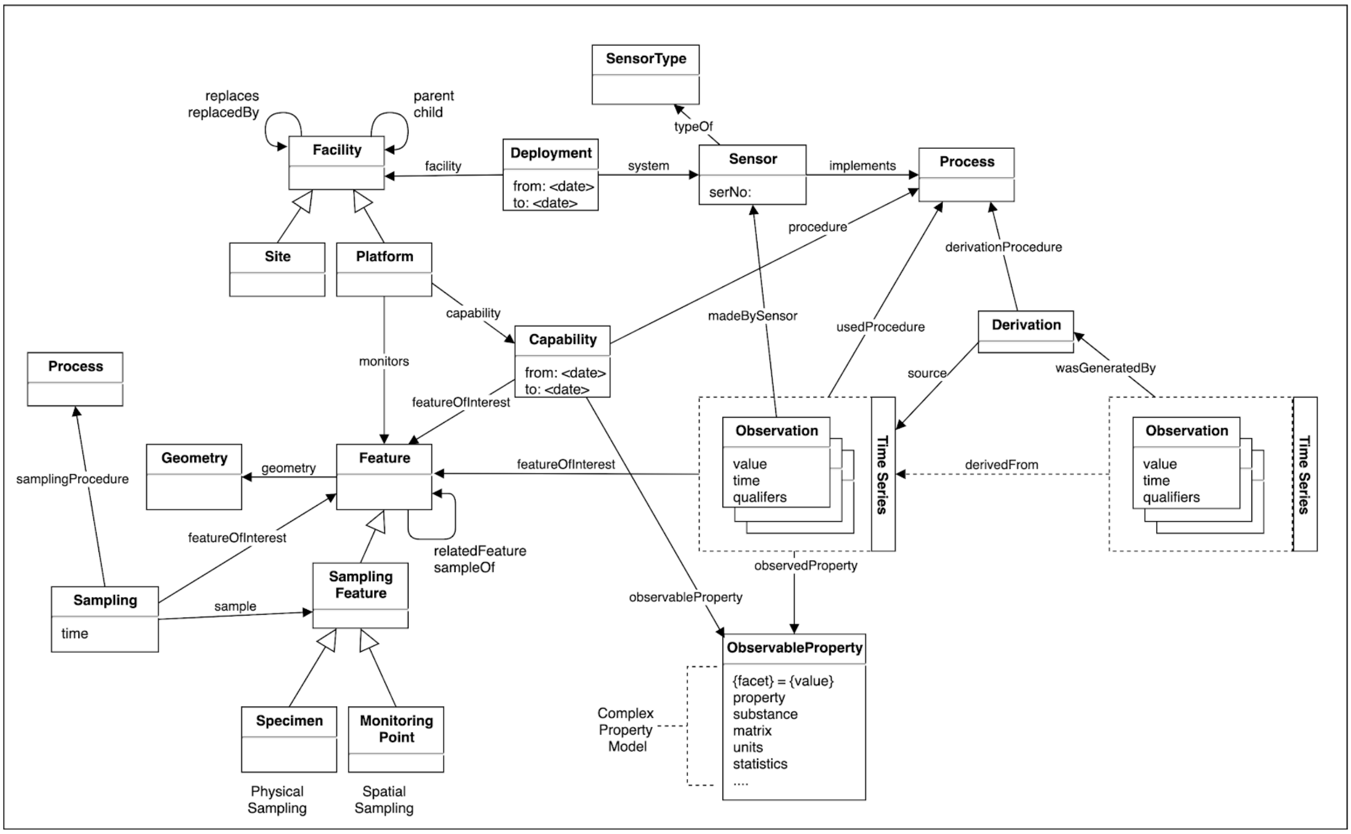


Figure 2 Conceptual Overview

At the heart of this structure are acts of observation and their results (readings) organised into time series. Observations estimate the value of some observable property of some feature-of-interest. In general, features are a somewhat generic entity intended in the main to cover the geospatial aspects of what is being observed, though in practice they can range very crisp point locations such as a monitoring points through to something more phenomenal such as River Pollution in the UK. To address the wide ranging nature of features (of-interest), features such as monitoring points and physical specimens collected at monitoring points, may be related to one another building up chains of related features such that some local or proximate feature-of-interest may serve as a proxy for some ultimate feature-of-interest.

Observation results (readings) are collected by sensors which may be thought of as implementing some process or procedure used in making an observation. Similarly, time series (and the observations from which they are made) may be derived from one another by some derivation process which may cleanse and/or aggregate data from a source time series using some given derivation procedure.

Sensors are of a given sensor type. Sensor types can also be thought of as a more generalised process that implements the generality of what the sensor does. Sensor types can be layered to express sensor make and model, modes of operation, as well as potentially a more general sense of the sensor technologies employed[4].

Sensors are deployed to facilities such as sites and platforms that host them for some period of time. Maintenance and calibration cycles may take a specific sensor out of service and replace it with a similar model which can be captured as deployment records. Facilities, such as sites and platforms are intended to cover or observe some feature or features of interest and as a consequence of being potential hosts for sensor deployments, they accrue observing capabilities in terms of the feature, observable property and procedure combinations they take from their location and sensor deployments.

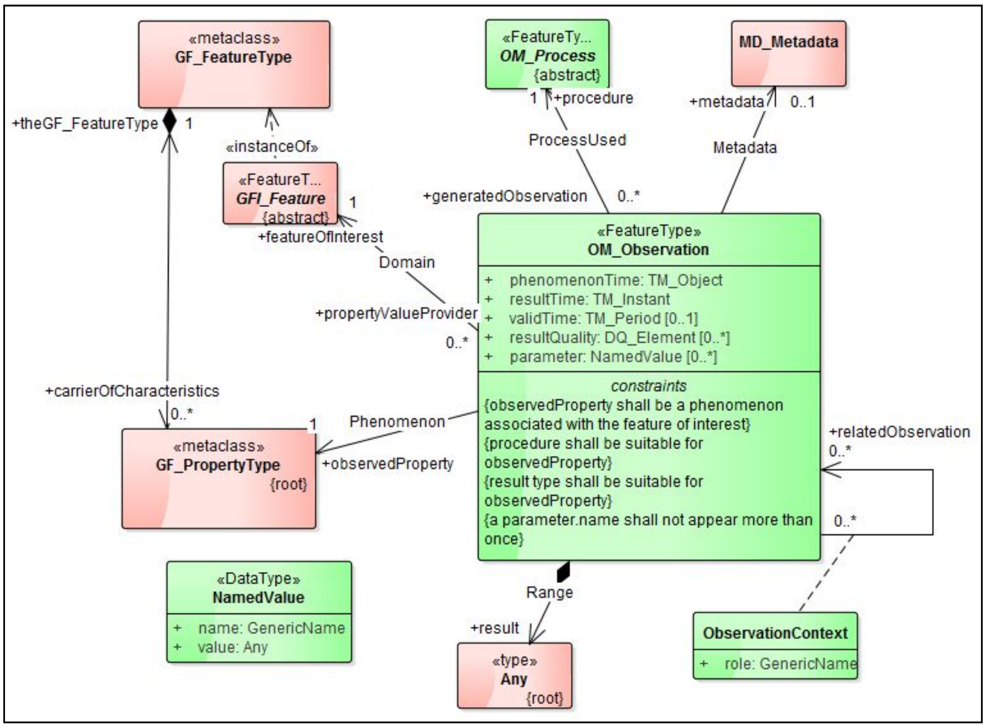
Sampling is the act of taking or selecting samples of some feature. Samples can then serve as a feature-of-interest for further observations. This applies equally to the selection of, say, monitoring points which are spatial samples of some larger feature and to the taking of physical specimens such water or rock samples taken from a particular location as part of a monitoring programme. In both cases, the samples, whether physical or spatial, become proximate features-of-interest in some observation process whilst they maintain a relationship to the feature that they are a sample of. The sampled feature may also prove to be the ultimate feature-of-interest of an observation, though in concept there may be several layers of sampling and sampling features, for example a deep water sample or an ice core held in an archive may be further sampled to provide specimens for further analysis - which may themselves be subdivided.

[4] https://cosmos.ceh.ac.uk/network-instruments: "Cosmic-Ray Soil Moisture Sensor" vs. specific type "Hydroinnova CRS-1000/B"

4.1 ISO 19156/OGC Topic 20 Observation and Measurements (O&M)

Focus: Representation of acts of Observation and Measurement, and their results.		
Primary Concepts	Observations, Measurements	Observations generally may have quantitative or categorical/classification results Measurements are observations with quantitative results.
	Feature of Interest Proximate Feature of Interest Ultimate Feature of Interest	Indicates the 'feature' being observed. Unfortunately, the nature of features ranges from purely geographic (the river Exe at Exebridge) to phenomenal (River Pollution in the UK). Often the immediate feature of interest (the proximate feature - e.g. a monitoring point) serves as a proxy for something more significant (the ultimate feature - e.g. the river segment in the vicinity of the monitoring point or indeed the whole river).
	ObservedProperty	The property of the feature-of-interest being estimated by the observation. A simple concept on the surface, but in practice it can lead to nuanced differences due to differences in observation procedures/instrumentation units of measure, aggregation methods, frequency of observation. As a concept, observable properties may be organised into hierarchical/thematic groupings.
	Procedure Used	The procedure/process used in making an observation. At a minimum a concept with a unique name and possible links to narrative articulation. Maximally there could be a machine readable articulation of a composite process.
	Sampling Feature eg. a monitoring point, a specimen	Sampling features are a category of features intended to serve as proximate features of interest that serve proxy for some more significant feature of which they be a spatial or physical sample of.

ISO/OGC Observations and Measurements (O&M), in common with many ISO191xx series specifications, provides an abstract conceptual model[2] expressed in UML along with a concrete realisation[3] expressed as an XML application schema. O&M provides a domain independent foundation representing observational information. The O&M glossary defines observation as the **act** of measuring or otherwise determining the value of a property". In particular the property being observed is understood to be an observable property of some feature-of-interest.



[Copyright © 2013 Open Geospatial Consortium](). See [2]

Figure 3 ISO 19156 Figure 2 - The basic Observation type

At its heart, O&M models an observation as a time value pair (**phenomenonTime, result)** with linkage to what was observed (**featureOfInterest**, **observedProperty**) and how the observation was made (**procedure**). Complexity arises in the factoring of aspects of 'what' and 'how' between **featureOfInterest**, **observedProperty**, **procedure** and **result,** see section 5 Cross-cutting Themes and Issues for further discussion. The observation **result** itself is open ended and may be a simple or a complex structure.

OM_Observations carry timing information in the form of:

- phenomenon time which indicates the time or time interval to which the result applies. In the context of forecast observations, phenomenon time may indicate a time in the future.
- result time which indicates the time at which a result became available. This enables a distinction between, for example, when a specimen was taken (phenomenon time) and the time at which a lab result arising from its analysis becomes available.
- and optional valid time which indicates the interval over which the result is intended to be used.

O&M is intended to be very generic and to be capable of recording direct, in-situ observations, sampled ex-situ observations and remote observations, for example satellite based earth observations which can lead to complex and diverse usage patterns.

4.2 W3C/OCG Semantics Sensor Networks/Sensors, Observations, Samples and Actuation (SSN/SOSA)

Focus: Broad... a core focussed on representing acts of observation, sampling and actuation, supported by structures for describing systems (senors, samplers, actuators) and the platforms that host them		
Primary Concepts	Observation, Features of Interest, Observable Properties, Processes/Procedures	As O&M
	Sampling, Sampler and Sample	A more explicit treatment of the act of sampling than O&M. An act of Sampling follows the same pattern as an act of Observation but with a Sample as its result.
	Actuation, Actuator, ActuatableProperty	Coverage of actuators as a dual of sensors.
	Systems, Platforms, and Deployments	Covers the deployment of systems (sensors, actuators, samplers) to platforms that host them. Platforms are part of some composite facility such as a site or monitoring station.

W3c/OGC SSN/SOSA[5] describes itself as a light-weight ontology for modelling acts of observation, sampling and actuation, using sensors, samplers and actuator respectively. With respect to acts of observation, SOSA share a similar if not identical conceptual basis as OGC/ISO O&M[2]. SSN/SOSA is formulated as an RDF vocabulary expressed in RDF/OWL

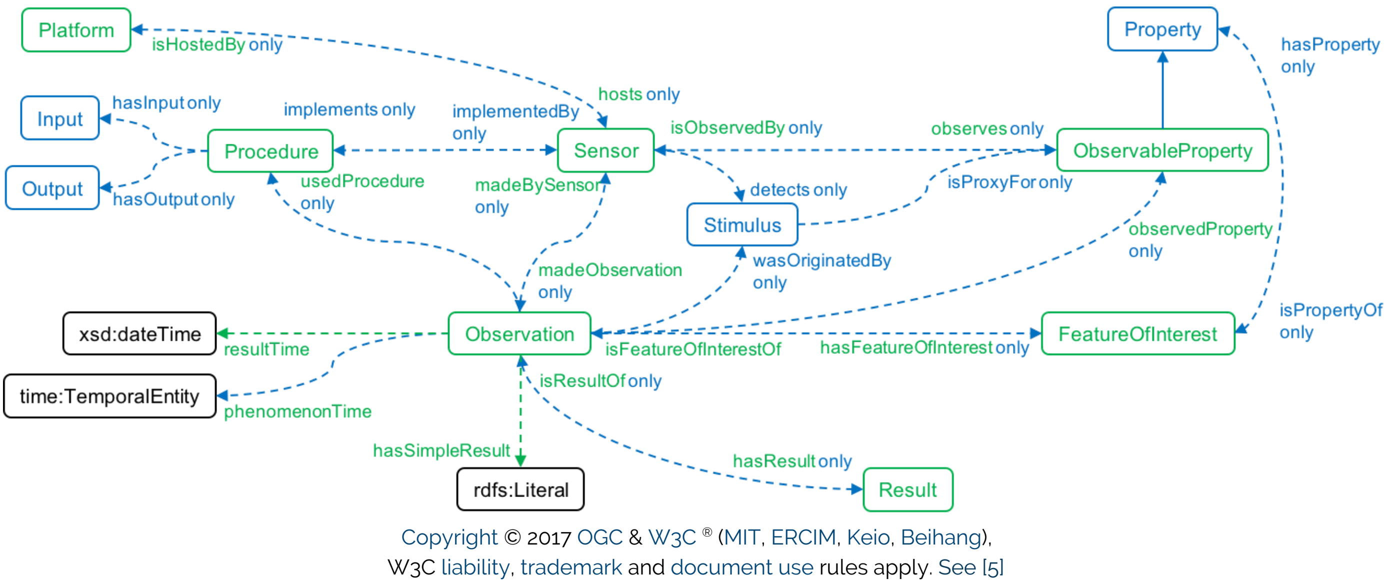


Figure 4 SSN/SOSA Observation (from W3C Recommendation)

It is evident that **resultTime**, **phenomenonTime**, **observedProperty**, **hasFeatureOfInterest, hasResult** (or **hasSimpleResult**) and **usedProcedure** are direct analogues of the corresponding O&M attribute and associations/association roles.

SSN/SOSA separates the observing sensor from the procedure used in making an observation - which differs from other practices, principally SensorML which regard sensors as processes that instantiate more generalised processes (c.f. procedures) with more of their configuration concretely bound (configured in a particular way, deployed into a particular situation). The observing capability of a sensor is indicated by the **observes** relation between a sensor and an observable property. Currently observed properties are captured in the Deployment entity (see figure 6 below).

Unlike O&M, which does not model the act of taking a sample, SSN/SOSA makes explicit provision for both acts of sampling and of actuation (i.e. acts that alter the state of the world). From a UKCEH perspective our focus is on observations and sampling.

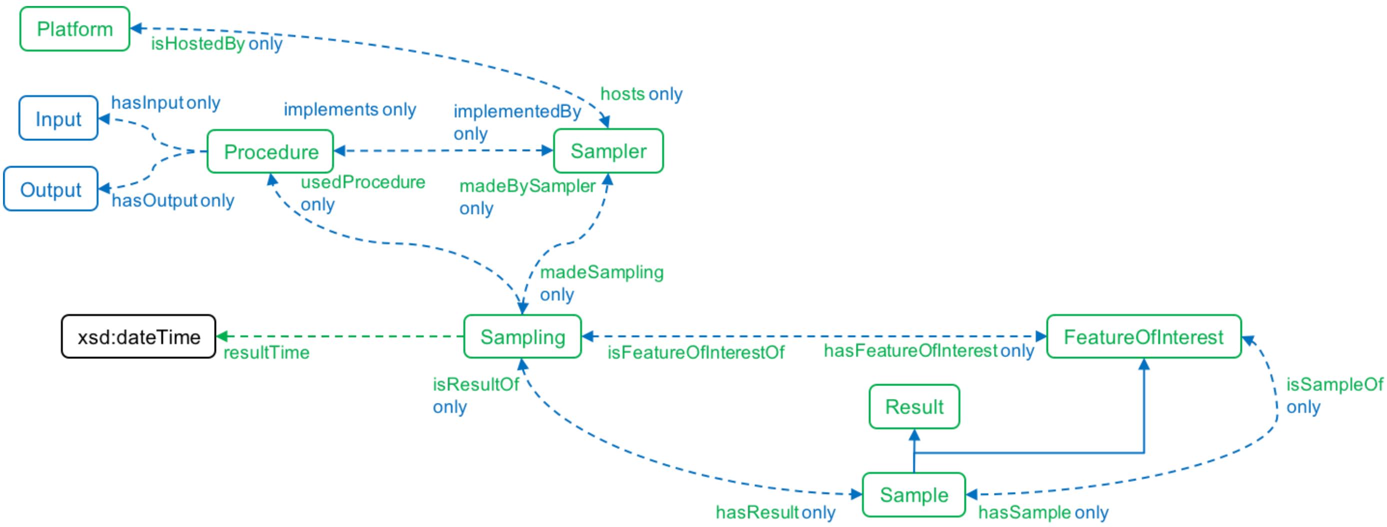


Figure 5 SSN/SOSA Sampling (from W3C Recommendation)

As with observations, an act of sampling has a result. In this case a sample of some feature-of-interest made by a sampler using a procedure and which has a **resultTime** indicating when the result (i.e sample) was made available.

SSN/SOSA regard sensors, actuator and samplers as systems (which may have composite structure via the **hasSubSystem** relation) that implement procedures and which are hosted by platforms. For example a COSMOS-UK monitoring site could be modelled as a platform that hosts a number of systems (soil sensors, rain gauges, wind speed gauges, temperature probes etc). The System model also enables expression of sensor capabilities (such as measurement precision) under different conditions (e.g. outer temperature).

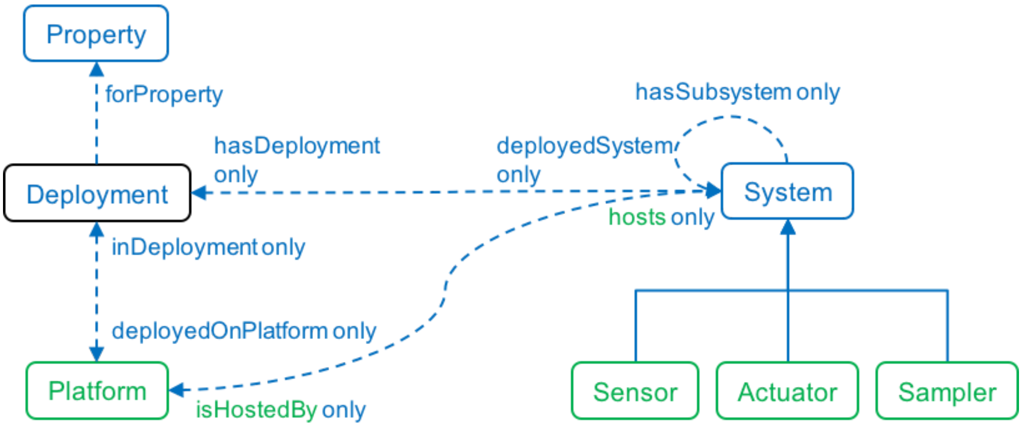


Figure 6 SSN/SOSA Systems (from W3C Recommendation)

Some UKCEH sensors may be deployed multiple times to different sites as maintenance and calibration processes recover sensor from the field and redeploy them to potentially different sites. SSN/SOSA deployments would need to be augmented with a time interval to capture the details of such time varying deployments. We would also need to develop some patterns to capture the distinction between a logical sensor (the air temperature sensor at a particular COSMOS-UK site) and the physical sensor (with a serial number) taking the measurement at a given time. For this, a distinction between a Platform and a Sensor could serve as a useful tool. We might also need to extend the model to capture links between a Platform and Features (of Interest) it is set to observe.

4.2.1 SSN Extensions (SSN-EXT)

The W3C are working on a set of extensions[6] to SSN/SOSA that adds two significant features. Note that at the time of writing SSN-EXT is a W3C working draft and has not (yet) achieved recommendation status.

- An observation property, **ssn-ext:hasUltimateFeatureOfInterest**, for designating the ultimate feature of interest
- A nestable **ssn-ext:ObservationCollection** class along with a membership property, **ssn-ext:hasMember.**

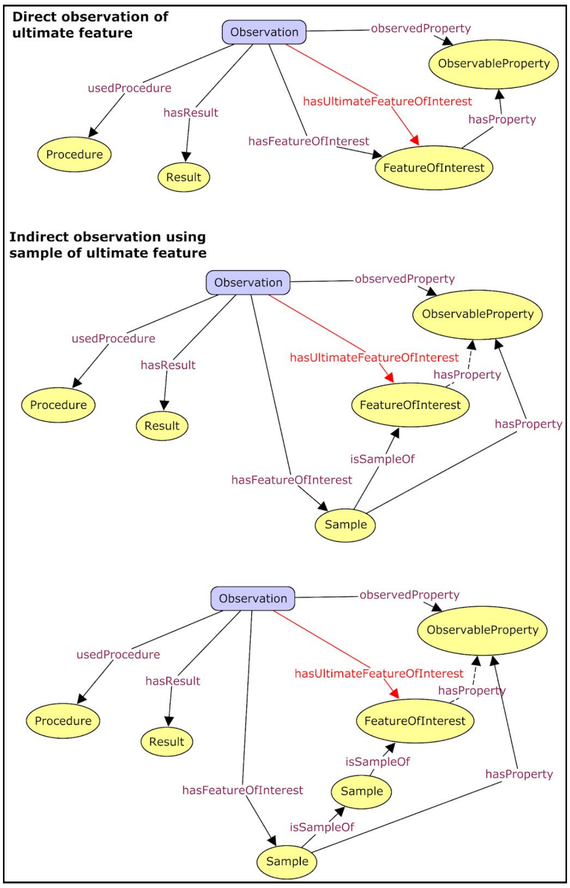


Figure 7 Patterns for observations that relate to the ultimate feature of interest directly (top), or indirectly via one sample (middle) or a chain of samples (bottom) - new property shown in red (from W3C Working Draft)

Figure 7 taken from the W3C working draft illustrates the use of **hasUltimateFeatureOfInterest** for both direct observation and two cases indirect observation through sampling. SSN-EXT also proposes a similar pattern for the act of sampling.

SSN-EXT Observation collections can be nested in that they may have both observations and observation collections as members. A number of property values common to a collection of observations can be carried on the observation collection, rather than repeated individually on each observation. This can occur recursively, so for example **observedProperty**, **usedProcedure** and **hasFeatureOfInterest** values may be carried by a top level observation collection while **madeBySensor** may be carried on a sub-collection and timing and result values may be carried on individual observation members.

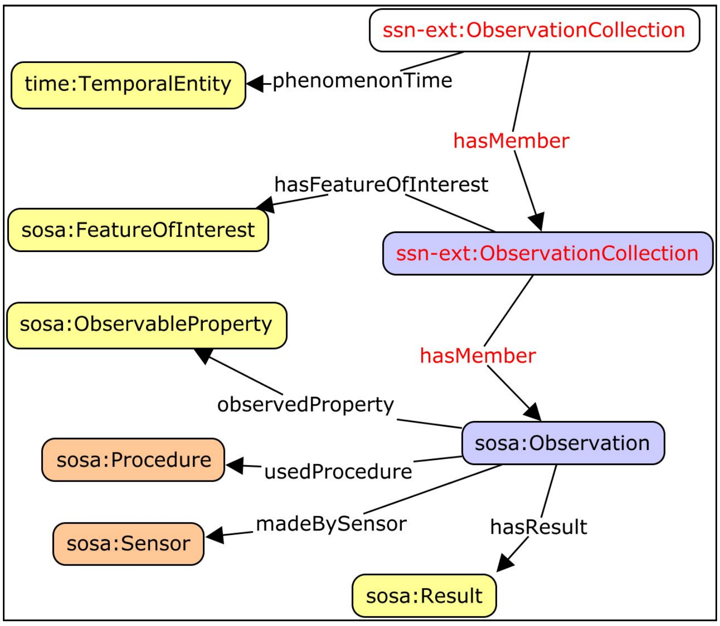


Figure 8 SSN Observation Collections (from W3C Working Draft)

The basic rule for SSN-EXT observation collections is that properties on an observation collection apply to all members of that collection, and recursively so.

A key limitation of SOSA/SSN (and O&M) for UKCEH purposes is that they have no explicit notion of a *time series*. Each **Observation** separately relates to a **Sensor**, **FeatureOfInterest** and **ObservableProperty**. This proposed *collections* extension has the potential to address this limitation. An Entrain ontology could define a *time series* to be a **ObservationCollection** and could then group all the metadata (about what property and feature were observed (and by what sensor) at the top level, the individual observations within the series then just need to indicate the time of the observation and the result. It would also allow grouping of time series into higher order collections, or splitting of time series into component sub-series (such as sub-series for each change of a sensor).

4.3 W3C Provenance Ontology (PROV-O)

<table>
<tr><td colspan="3">Focus: The derivation of entities from one another through the action of activities involving agents that may be acting on behalf of other agents. W3C/OGC SSN/SOSA provides an alignment module that maps many of its concepts and relations to the concepts introduced by PROV-O</td></tr>
<tr><td rowspan="3">Primary Concepts</td><td>Activity</td><td>Observations, Samplings, Actuations (SOSA) and Derivations are all forms of activity.</td></tr>
<tr><td>Entity</td><td>More or less anything that can be generated by activity or used by an activity to generate (create) new entities. This would include 'raw' materials and also specifications that guide the activity.

Features, Samples, Observation results, Observed properties, process specifications and so-forth are all entities that may be produced or generated by activity.</td></tr>
<tr><td>Agent</td><td>An active participant in an activity or on whose behalf an activity is conducted.Agent includes people and organisations as well as equipment such as sensors, sampler and computational equipment that may perform computed derivations.</td></tr>
</table>

The W3C Provenance Ontology (PROV-O)[7] provides a model for expressing the provenance of entities based on three main classes and nine primary relationships illustrated in figure 9 below

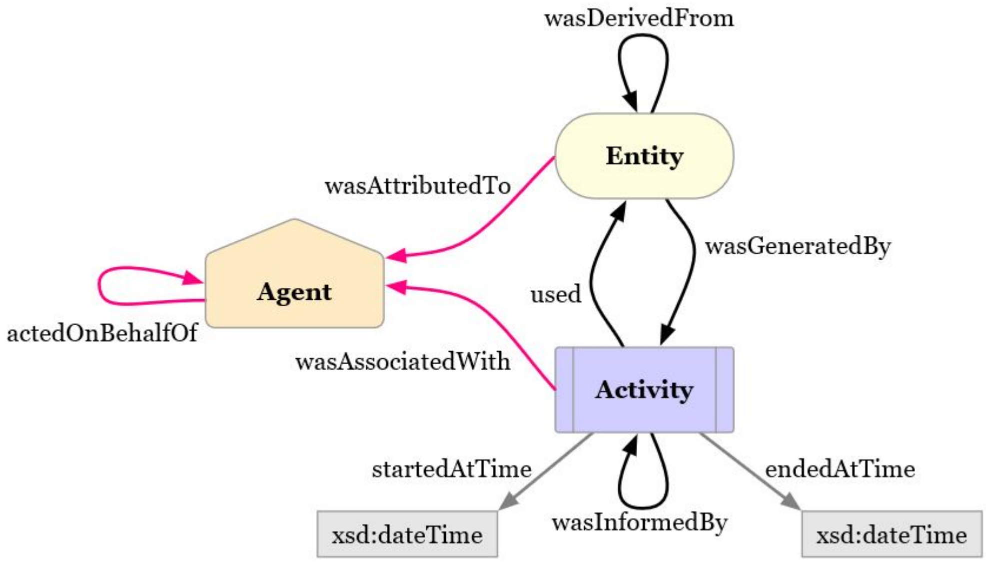


Figure 9 W3C PROV-O core classes and relations (from W3C spec).

The basic model is that activity involving agents (people, organisations, automata…) uses entities (things… more or less anything) to make more entities. In this way entities are derived from one another and attributed to the agents involved in the activity that created them. The activity itself occurs over some interval of time.

Beyond these basic relations, PROV-O provides qualified variants that effectively enable instances of these primary relations to carry further details. For example a **prov:Generation** may be used to indicate when, during the course of an activity, a given entity was actually generated:

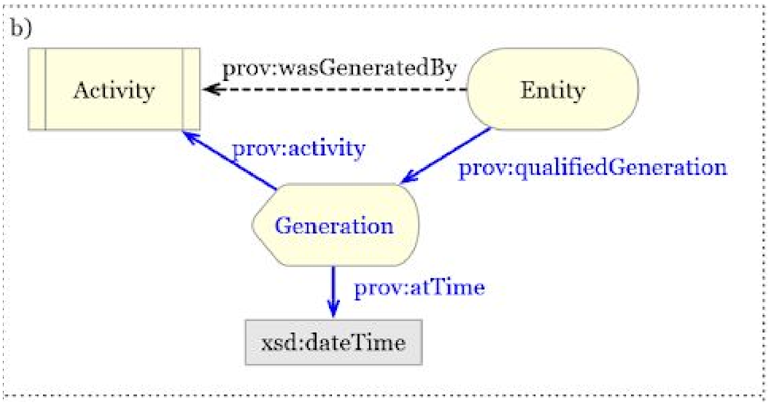


Figure 10 Qualified variant of wasGeneratedBy

Given the somewhat recursive nature of PROV-O with entities derived from entities, it is clearly possible to generate considerable provenance information for a single activity. There is a need to be pragmatic with respect how far to follow the derivation of an entity into the past.

For UKCEH observational data, it is important to capture the details of the instrument and procedures used in making an observation sufficient to correctly interpret the observation result. The PROV-O alignment module of SSN/SOSA[5] aligns **sosa:Observation** as a subclass of **prov:Activity; sosa:Sensor** as a subclass of **prov:Entity** and **prov:Agent;** and **sosa:isResultOf** as a sub-property of **prov:wasGeneratedBy**. In this way the use of SSN/SOSA in describing an observation can also be seen as making provenance statement about the generation of the observation result.

4.4 OGC Time-Series ML

Focus: The representation of time-series as a one-dimensional coverage - a function from time to value. In particular in the context of observations and measurements, Timeseries ML frames a time series as the result of a single observation.		
Primary Concepts	Timeseries	A one-dimensional coverage from time to value with metadata (e.g. aggregation functions and qualifiers) carried both by the time series as a whole and by individual time-value points (c.f. observation collections and observations). Timeseries ML presents two time series encoding options, either as 1) a domain (from) and range (to) mapping (domain-range) or 2) as a collection of time-value pairs (TVP).
	TimeseriesObservation	An O&M Observation with a time series result.

[5] https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/#PROV_Alignment

OGC Timeseries ML[9] is titled “Timeseries Profile of Observations and Measurements” and is derived from an earlier work, “OGC WaterML part 1 - Timeseries”[10].

Timeseries ML notes two ways of framing a time series as OM_Observations:

1. *A collection of OM_Observations. Each observation represents a single data point; the collection makes up a timeseries.*
2. *An OM_Observation whose result is a discrete coverage that varies in time (c.f. OM_DiscreteCoverageObservation). Here the OM_Observation feature type provides the spatio-temporal context for the whole timeseries.*

Timeseries ML focuses exclusively in the second of these alternatives, i.e. framing a time-series as one-dimensional coverage observation result. Bare use of **TimeSeries::TimeSeries** (see figure 11 below), without being framed as an observation result, is potentially very close to existing UKCEH data structures and intended usage patterns.

Timeseries ML is useful in situations where an individual observation result needs to capture property variation over some bounded interval of time - for example where the observed property is say daily cumulative rainfall sampled on an hourly basis.

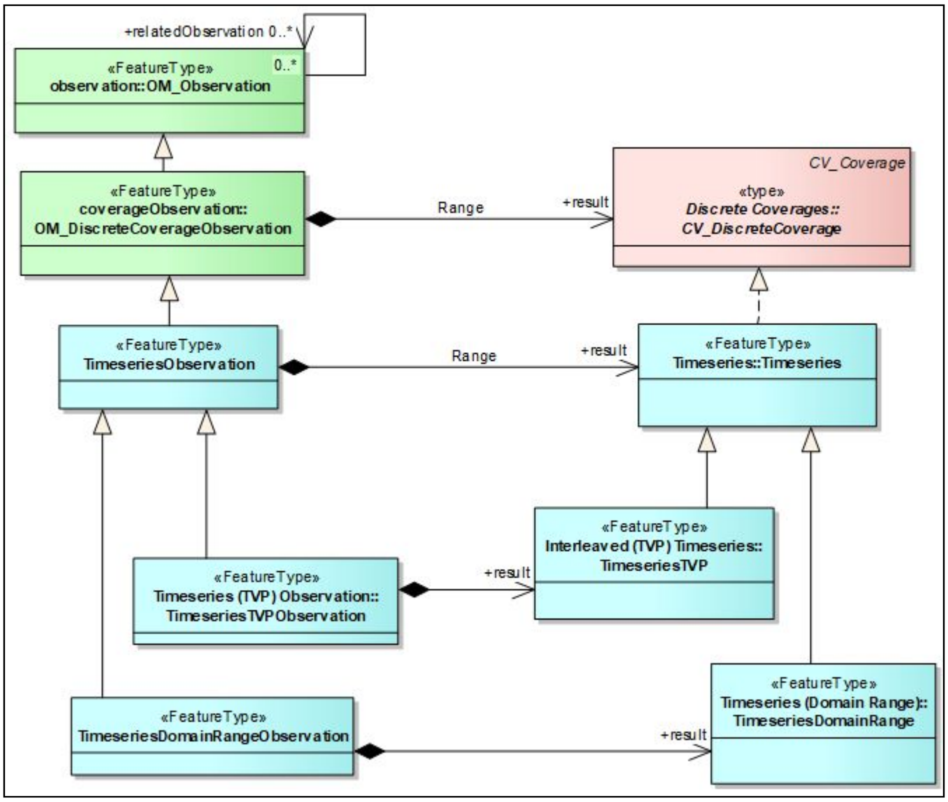


Figure 11 Timeseries ML: Time-series as specialised observations

Timeseries ML provides for two different encodings of time-series, either as :

1. Domain-Range encoding where the domain (time axis values) and range (value axis values) are given as separate collections, each potentially with their own associated metadata, or;
2. Time-Value Pair (TVP) encoding where each time-value pair in the time series is given separately, and each TVP could carry its own metadata.

In either case metadata may be supplied for the time-series as a whole and for each point in the series.

Timeseries ML provides a number of useful code list for encoding:

- interpolation approach between adjacent points
- data quality
- type of observation procedure

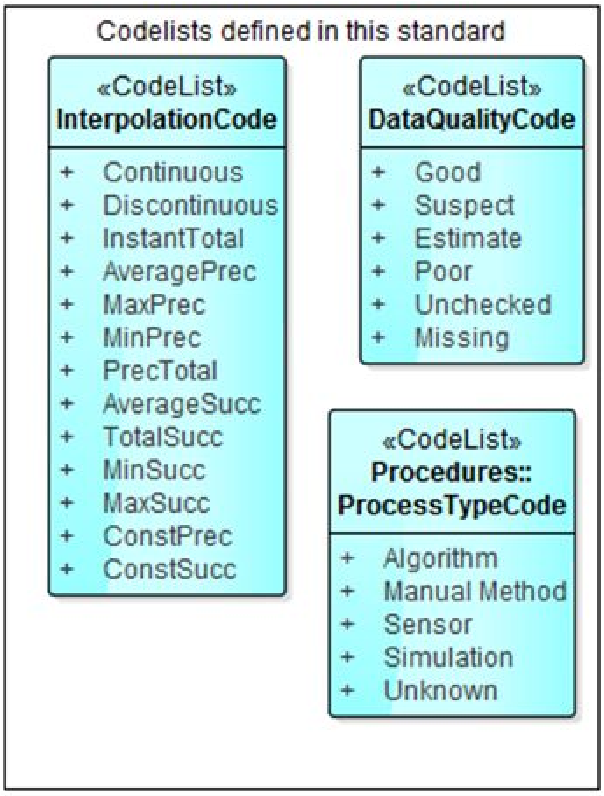


Figure 12 Timeseries ML codelists

4.5 INSPIRE Environmental Monitoring Facilities (EMF)

Focus: Hierarchical arrangements of monitoring facilities, their lineage and monitoring capabilities. Also the arrangement of monitoring facilities into networks and the programmes that they support,		
Primary Concepts	Environmental Monitoring Network	Collections of multiple facilities that form monitoring networks. Facilities can be members of multiple networks and network memberships may be time bounded.
	Environmental Monitoring Facility	Environmental monitoring facilities may be composite and built up from multiple subordinate facilities. Typically specialised as sites or monitoring stations composed of platform that host sensors. Conceptually sensors, samplers and actuators can also be regarded as facilities, however they are more transient due to maintenance and calibration activities.
	Observing Capability	Facilities accrue capabilities to observe properties of particular features using particular observation methods or procedures through the deployment of sensors to facilities.

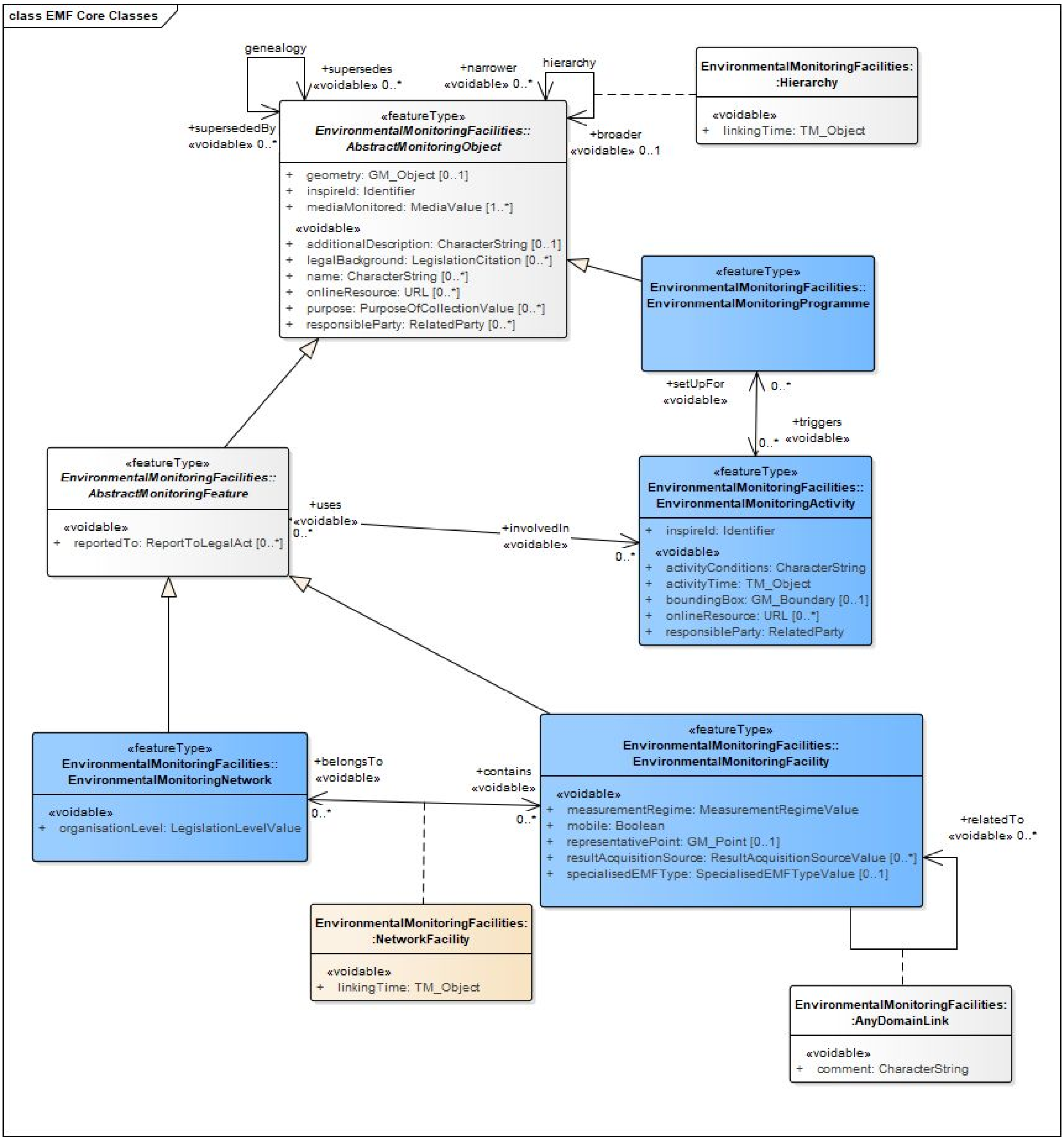
The INSPIRE Environmental Monitoring Facilities (EMF) Theme Data Specification [14] provides data models that cover:

1. the description of environmental monitoring networks, facilities, programmes and activities.
2. the observational capabilities of networks, facilities and programmes.

EMF introduces 4 principal spatial object types (synonymous with ISO 191xx feature types): Environmental Monitoring **Facility**, **Network**, **Activity** and **Programme** as illustrated in figure 13 below:

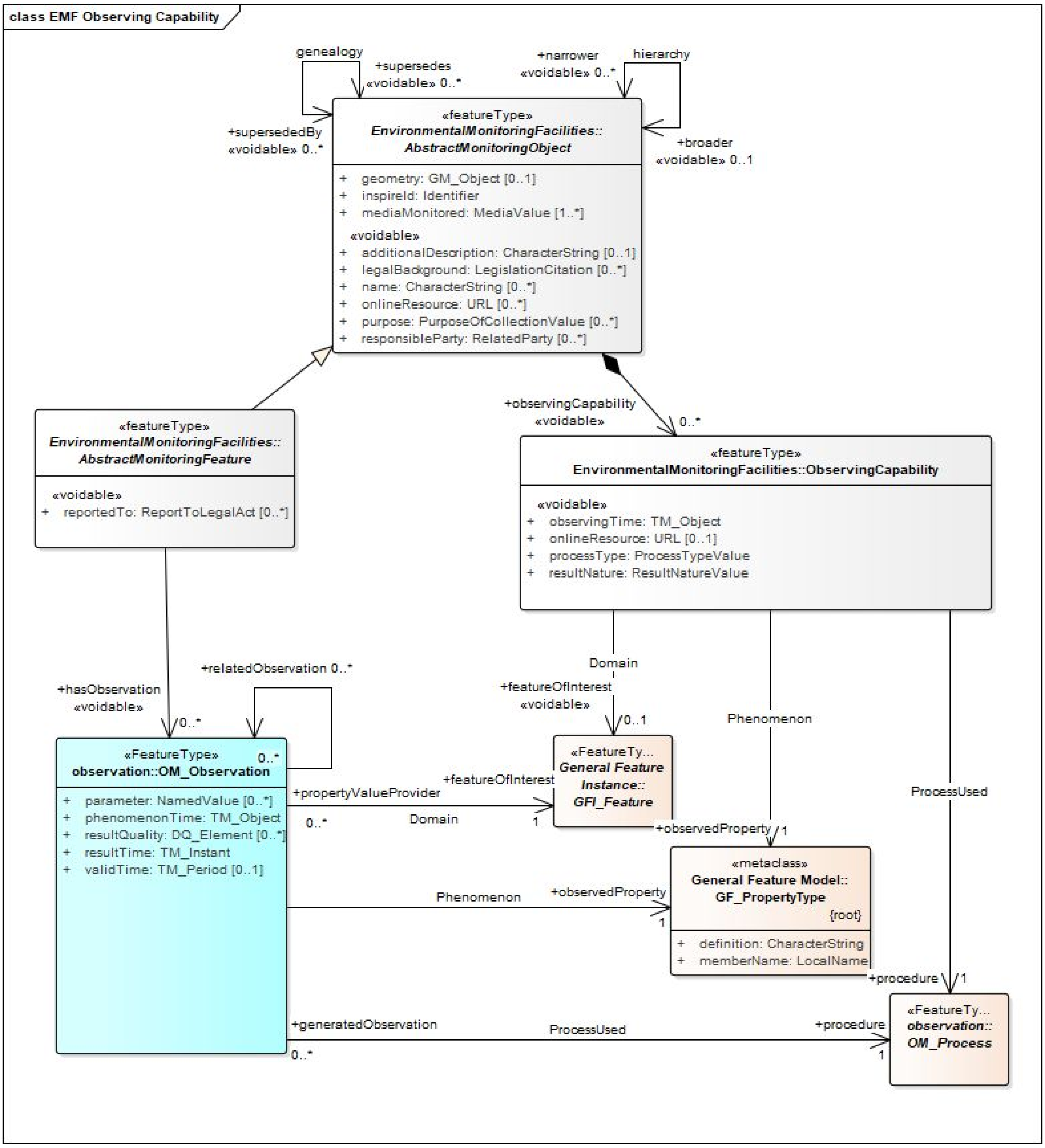
The class **AbstractMonitoringObject** provides common properties for indicating succession (lineage) and composite hierarchy so that programmes, networks and facilities can be built from smaller, replaceable, parts. The hierarchy relation between facilities can be used to implement a site, platform, device, sensor hierarchy. The relationships between networks and facilities and between facilities (**relatedTo** and **hierarchy**) are accompanied by linking times or intervals over which the corresponding associations hold. These can be used to model, say, the re-deployment of a sensor or device from one site to another.

To address the observational capability of an environmental monitoring facility, INSPIRE EMF, introduces the **ObservingCapability** class (see figure 14) which in a sense serves as a template for the observations generated by the facility in that it indicates the associated **featureOfInterest**, **observedProperty** and **procedure** values. The presence or use of observing capability makes it possible to search for facilities based on aspects of their capability.



Published by European Commission Joint Research Centre [14]

Figure 13 EMF Core classes and properties



Published by European Commission Joint Research Centre [14]

Figure 14 INSPIRE EMF Observing Capability

4.6 OGC SensorML

<table>
<tr><td colspan="3">Focus: The description of sensors (instances), sensor types (categories) and their deployments as processes. Sensor ML also covers the description of non-physical processes such as computational processes for cleansing and/or aggregating sensor data. Sensor ML also covers the shape and configuration of data flows between process elements in an aggregate process.</td></tr>
<tr><td rowspan="4">Primary Concepts</td><td>AbstractProcess</td><td>All Sensor ML processes are derived from AbstractProcess. It provides for the definition of process inputs and outputs; association with features-of-interest; semantic grounding (definition); configuration; and specialisation (typeOf relationships).</td></tr>
<tr><td>AggregateProcess</td><td>Provides for the composition of aggregate processes from other processes (components) and the interconnection of their inputs and outputs,</td></tr>
<tr><td>PhysicalProcess</td><td>Processes that are implemented by physical components such as sensors.</td></tr>
<tr><td>NonPhysicalProcess</td><td>Computational processes that transform their inputs into outputs. These may themselves be composite or simple.</td></tr>
</table>

OGC SensorML 2.0 [17] provides a data model for describing sensors, actuators and processes in general. The latter is particularly useful for describing data clean-up, aggregation and/or transformational processes on primary data source.

SensorML models sensors and systems (inc. actuators) as physical processes which may be subjected to repeated execution, giving rise to discrete observations, or which may be on-going, giving rise to streams of data. The core of SensorML provides for the description of aggregate processes and the interconnection of data flows between processes. Component processes may be physical, non-physical or indeed other aggregate processes.

The shape of a sensor's output may be specified as Sensor Web Enablement (SWE) data components using the SWE Common Data Model [18]. Sensor outputs can be packaged as O&M observation results or transferred via a data stream. The SWE Common Data Model provides basic simple and range data-types and the means to create more complex records from these basic types. The semantic significance of a data component can be grounded using the **definition** attribute carry a link to an online dictionary or register of definitions.

When packaging Sensor outputs as O&M observation results, SensorML advocates that the procedure used to make the observation be described as a SensorML process. This makes most sense if the observing process models the sensor that generated the result. The sensor process (an **sml::PhysicalProcess**) may itself be derived (**sml::typeOf** association) from a more abstract physical process (**sml::AbstractPhysicalProcess**) that has fewer of its configuration/deployment parameters bound and which can be thought of as representing the sensor type - or some partial configuration of the sensor that might be common to multiple installations/deployment (particular firmware revisions or operating mode settings).

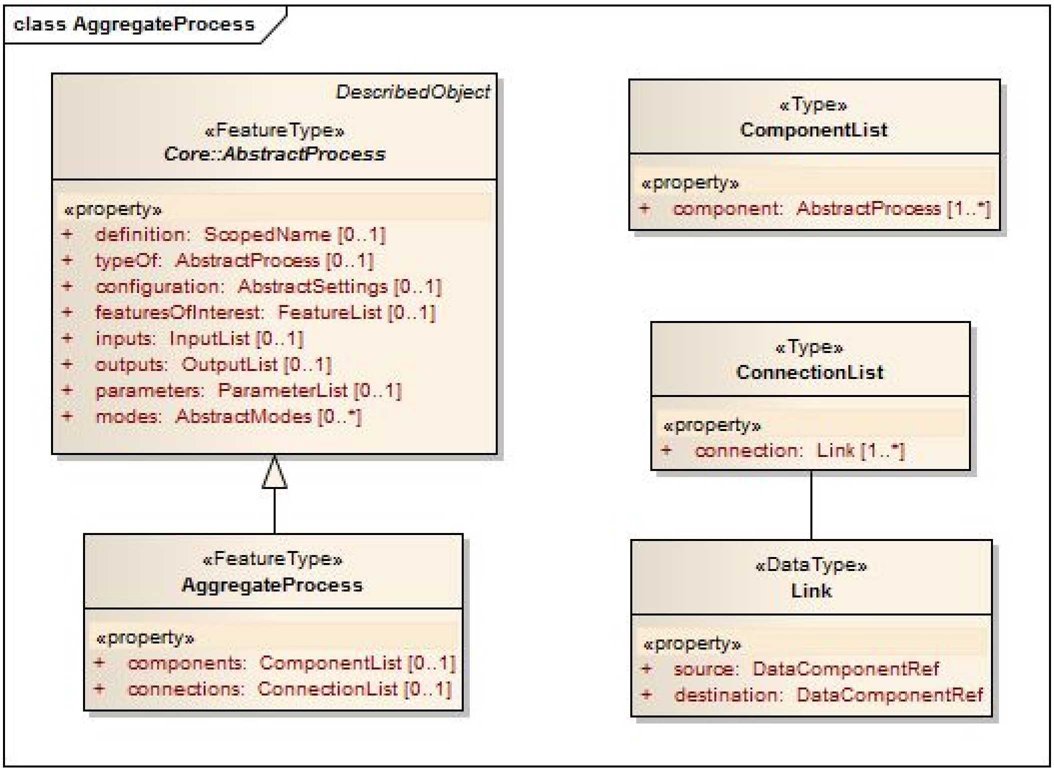


Figure 15 SensorML Aggregate Processes

SensorML enables descriptions of the features and properties observed by particular sensors, systems or their types, as well as description of result precision, sampling period and other capabilities.

4.7 OGC Topic 6/ISO 19123 Coverages

Focus: Mapping functions from some n-dimensional space, usually geospatial and/or temporal to value. Mappings may be discrete or continuous (usually a discrete mapping supported with interpolation rules).		
Primary Concepts	Coverage	A functional mapping from some n-dimensional domain to a range of values
	Domain	The 'from' side of a functional mapping, Domains can range from simple, eg. a single time dimension, or a simple x,y or x,y,z spatial dimensions to more complex x,y,z,t space/time dimensions together with associated coordinate reference systems for resolving spatial references.
	Range	Simple or complex structures representing the value of some property that the coverage represents at the corresponding domain values.

ISO 19123/OGC Topic 6 [19] provides a conceptual model for coverages. An ISO 19123 coverage is a mapping from an n-dimensional spatio-temporal (domain, typically 2D-4D) to value (range). As with an O&M observation result the typing of coverage range values is deliberately open-ended.

The coverage domain is made up of spatio-temporal objects that occupy some 'volume' of space/time. The specification defines operations for mapping from direct positions within the coverage domain's spatio-temporal envelope to values within the coverage's range.

The specification provides for a number of different approaches for representing discrete and continuous coverages.

Coverages can be used as observation results whose domain can describe the spatial coverage of a feature-of-interest and/or the temporal coverage of phenomenon time. For example earth observation by satellite typically generates imagery that covers some spatial extent on the ground. However, while an observation's feature-of-interest typically provides connection to its spatial element, in the case of an observation with a spatial coverage result, the coverage's domain conveys the spatial aspects of the observation's feature-of-interest.

4.8 W3C Data Cube Vocabulary

Focus: Collections of observations organised around dimensions, attributes and measures. Similar in concept to OGC coverages, but with origins in the statistical rather than geospatial communities.		
Primary Concepts	Dataset	A collection of data cube observations and associated slices, described by a data set definition.
	Observation	A data point situated in an n-dimensional space that can carry one-or-more measures and optionally multiple attributes that may qualify the interpretation of measure values.
	Dimension	The coordinate axes of an observations in an n-dimensional space (c.f. coverage domain)
	Attribute	Attributes or qualifiers associated with an observations that aid in the interpretation of the observations measures.
	Measure	One or more measures (c.f. observable properties) whose values are situated in the n-dimensional space at a position indicated by an observation's dimension values.
	Slice	Sub-groupings of a cube Dataset that share one or more dimension value in common. This is useful as a structure for optimising access to data cube values and may also be useful for grouping values with reduced dimensionality for presentation or aggregation purposes.

The W3C Data Cube Vocabulary [20] is an RDF vocabulary for describing the shape of n-dimensional observational data cubes. It has its origins in SMDX from the statistics reporting community rather than the geospatial OGC community. It provides a model for mapping from an n-dimensional space formed by a set of dimensions to values known as measures and accompanying attributes that affect the interpretation of the associated measures (e.g. units of measurement):

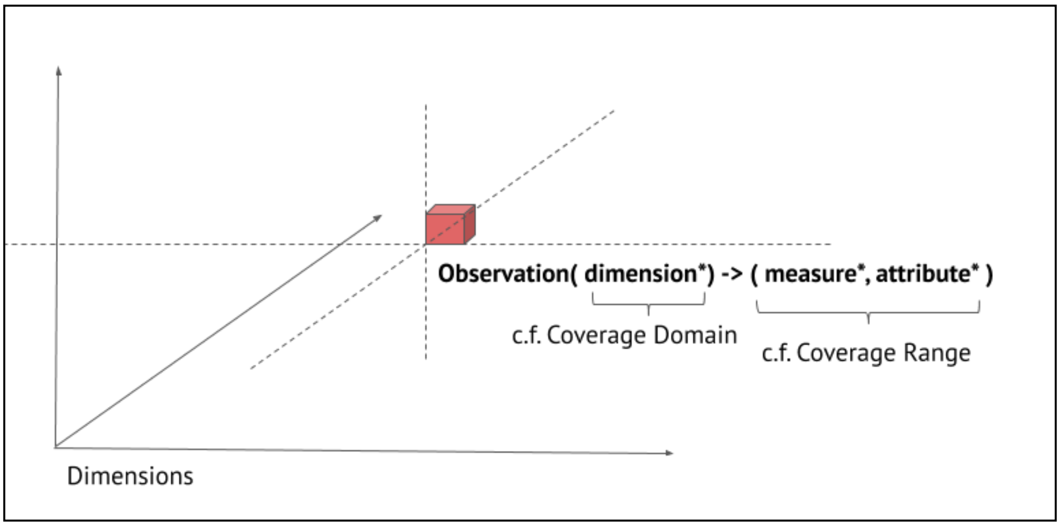


Figure 16 Data Cube Observations

The W3C Data cubes specification itself does not have the same deep connection between its dimensions and the spatio-temporal domain that the ISO 19123 coverage specification [19] has. However, it has been used in very similar contexts to O&M observations where monitoring points (effectively features of interest), determinands (c.f. observed property) and sampling time (c.f. phenomenon time) have been framed as cube dimension and the resulting measures and attributes as observation results.

The Data Cube vocabulary allows a set of observations to be grouped into slices and allows attributes (metadata) to be attached at the level of a slice or a whole data cube - to avoid repeating them for each observation. This is similar in spirit to the *collections* approach being developed in SSN-EXT. A slice can represent a time series at some location (feature of interest) and metadata (e.g. on the sensor) can be attached to the slice or a whole data cube. However, Data Cube does not natively support arbitrary collections of observations - a slice is defined by fixing one or more dimensions.

4.9 OGC Sensor Observation Service 2.0 Hydrology Profile

Focus: This profile focuses on the application of OGC Sensor Observation Services to the hydrology domain. SOS generally provides services for the storage and retrieval of sensor readings and sensor metadata. Of particular interest is their approach to time series identification. SOS generally makes use of both Timeseries ML and Sensor ML in formulating service responses.		
Primary Concepts	Time series identity	SOS 2.0 Hydrology Profile formulates time series identity as a function of feature-of-interest, observed property and observation procedure. Observation procedure is intended to indicate the type of sensor involved in making the observation. For UKCEH this may be inadequate because several sensors of the same type may be deployed to cover nominally the 'same' feature.

The SOS Hydrology profile, adopted as "OGC Sensor Observation Service 2.0 Hydrology Profile" [21] makes use of WaterML 2.0, in particular WaterML 2.0 part 1 Timeseries from which TimeseriesML is derived.

Within this profile a time series is identified by a unique combination of feature-of-interest, observed property and observation procedure (which acts as a proxy for sensor type, **not** sensor instance). In this context the time-series is on-going and of indefinite length.

Within the context of an SOS installation, at any point in time the time-series data held will be of finite length. The SOS 2.0 services that retrieve some or all of a time-series' data can wrap the response as an **OM_Observation** with a time series result. For time-series observations, **phenomenonTime** indicates the temporal extent of the observation result[6]. Similarly, the observation's **resultTime** is intended to convey when that particular result became available rather than the time at which the response was made available i.e. it will most likely match the most recent time point in the observation result. In this way the **OM_Observation** becomes a 'container' for the ongoing time-series which is produced by a continuous on-going process.

WaterML 2.0 prescribes that a sampling feature is used as an observations' feature-of-interest. The SOS Hydrology profile goes further and restricts that further to being a WaterML 2.0 monitoring point.

This profile is particularly interesting in the way that it frames the identification of a times series as described above:

ts_id = func(foi, op, proc)

And in particular in the case of physical sensors **proc** represents the sensor type rather than the individual sensor. This has two consequences that may be relevant to UKCEH:

1. A change of sensor type (different model, different firmware) may result in a change of procedure which would result in an undesirable change of time-series identity.
2. There is no direct association with the particular sensor (model/serial number/configuration) that recorded a given time point value.

Basing the (on-going) time series identity on just the feature-of-interest (monitoring point -> (site x sensor role)) and observed property would eliminate the first problem if an instrument is upgraded.

Sensor deployment records detailing the intervals during which a particular sensor is deployed into a given role at a given site would allow an association to be formed between a time point value and the sensor that recorded it. These associations could be materialised into individual time-point metadata. Or alternatively the time-point metadata could be expanded to include direct reference to the recording sensor. It is likely that lack of precision of time recording with respect to the retirement and deployment of a sensor may lead to misattribution of a few samples around the time of a change if deployment records are the sole basis of the attribution.

---

[6] It is not clear whether this in intended to represent the full extent of the time series or just the fragment included in the service response.

5 Cross-cutting Themes and Issues

The specifications reviewed in the previous section offer a great deal of flexibility. However, that means that they can be applied in different ways in different data collection and management situations. Typically flexibility in specifications is addressed by communities developing *profiles* for the way in which they are used by that community. There are a number of overlapping aspects that are covered differently by different specifications. In this section we consider a number of cross-cutting themes and issues that arise either due to flexibility in the underlying specifications or due to their differing approaches in addressing the same functional needs or requirements.

5.1 Observed/Observable Properties and Features of Interest

The **observedProperty** and **featureOfInterest** attributes of an observation that pervade the OGC/ISO 19xxx specifications and the observational aspects of SSN/SOSA are primarily intended to establish '*what'* it is that has been observed - '*what*' observable property of '*what*' feature. However, there are a number of *what* and *how* facets that are sometimes factored differently across feature-of-interest, observed property and procedure ('how').

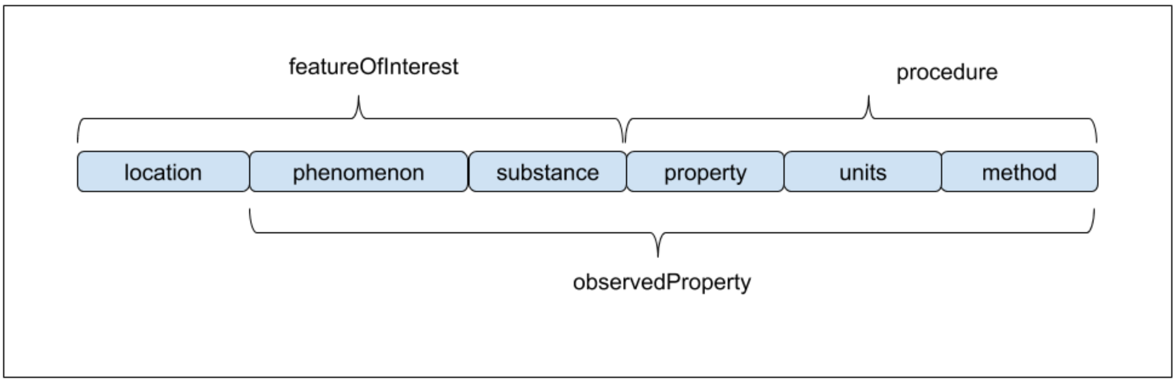


Figure 17 Observational facets and their factoring across observation attributes

For example in an environmental monitoring situation it is appealing to express a determinand as an observed property and to express a monitoring point (a sampling feature in O&M terminology) or the monitored feature itself as the feature-of-interest for the corresponding observation. However, when considering a determinand, it may also entail both units of measure (an aspect of the observation result) and method used in making the measurement. For example the table below includes some example observed properties from the NERC vocabulary server that measure the absorbance of a particular wavelength of EM radiation by different methods, and from the EA water quality archive linked data service that measure the concentration of 2,4-Dichlorophenol expressed in different units.

Observed Property	**Description/Definition**
SDN:P01::ABSP534A	Absorbance of electromagnetic radiation (534nm wavelength) {light absorbance} by the water body [dissolved plus reactive particulate phase] by ***spectrophotometry***

SDN:P01::SWLA534C	Absorbance of electromagnetic radiation (534nm wavelength) {light absorbance} in the water body [dissolved plus reactive particulate <0.1um phase] by ***filtration and spectrophotometry***
SDN:P01::PIGNA534	Absorbance of electromagnetic radiation (534nm wavelength) {light absorbance} by the suspended particulate material >GF/F [pigment phase] by ***filtration and spectrophotometry on residue before and after NaOCl bleaching and differencing of the measurements***
SDN:P01::SULAD534	Absorbance of electromagnetic radiation (534nm wavelength) {light absorbance} by the suspended particulate material >GF/F [non-algal particle phase] by ***filtration and spectrophotometry on residue after NaOCl bleaching***
EA:WIMS:0994	2,4-Dichlorophenol (concentration expressed in mg/l)
EA:WIMS:7997 and EA:WIMS:9816	2,4-Dichlorophenol (concentration express in ng/l)

Table 1 **Example observed properties and determinands that entail method and/or units measure**

Indeed EA:WIMS:7997 and EA:WIMS:9816 appear to be identical determinands that may have arisen due to a failure to find and reuse a pre-existing determinand definition. Alternatively there may be some invisible nuanced differences in the situations in which they are used.

The point being made here is that the clear separation between 'what' property of 'what' feature and 'how' observed implied by the **observedProperty**, **featureOfInterest** and **procedure** attributes of an observation does not necessarily arise in practice where facets of the phenomenon, method and units may be entailed in observed property and similarly facets of phenomenon and substance may be entailed in feature-of-interest (e.g. soil-moisture in the vicinity of a monitoring point).

Leadbetter and Vodden[1] recognised this problem and have developed the notion of a Complex Property Model where the properties themselves entail values for a number of facets. The property facets that the authors call out are:

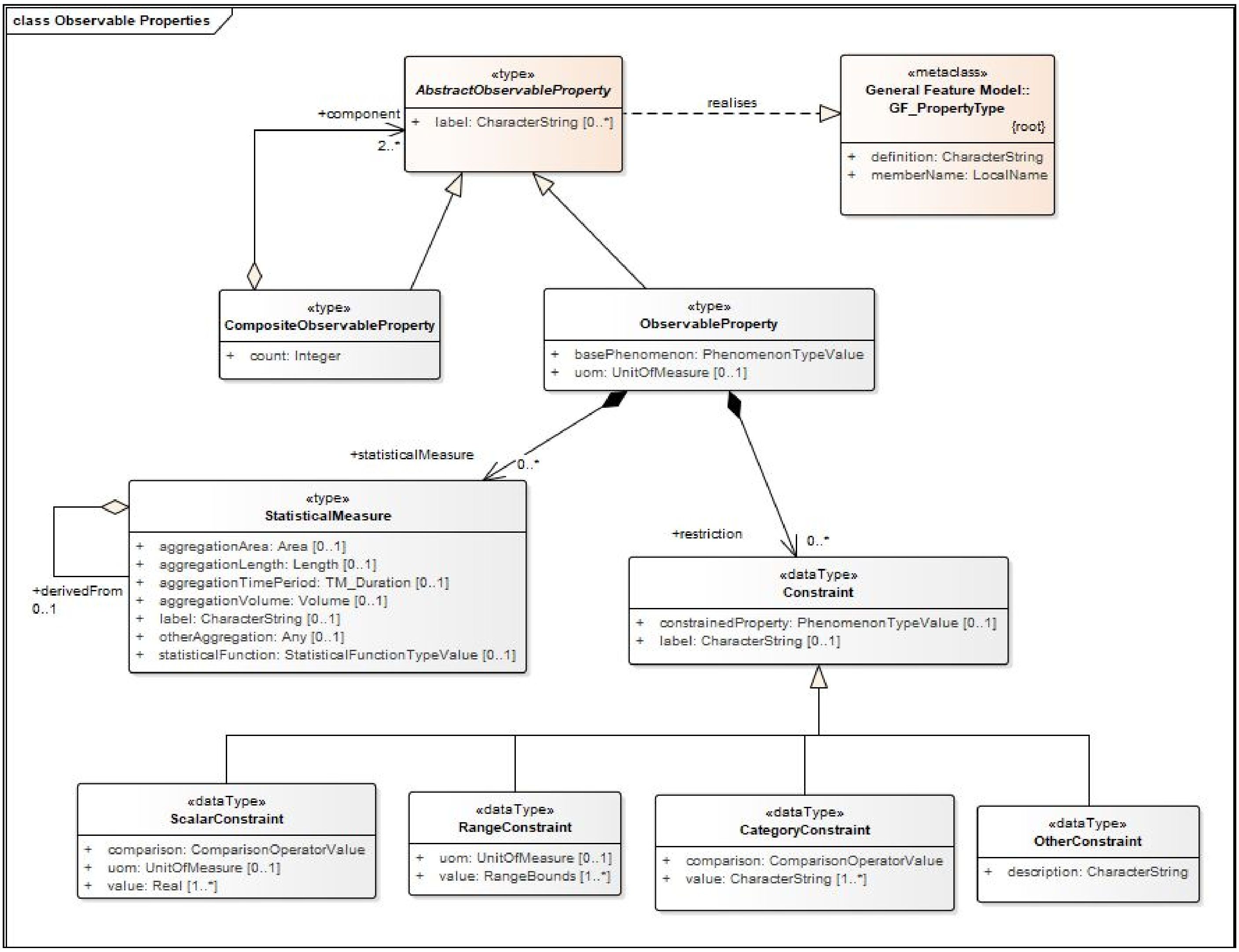
Facet	**Description**
ObjectOfInterest	Indicates more specifically what is being assessed by an observation. E.g. the substance whose concentration is being measured - eg. phosphate, lead…
Property	The generic property being measured: weight, concentration, length….
Unit Of Measure	The unit of measure associated with observations of this observable property (optional). QUDT serves are a good source of units of measure.
Statistical Measure	The statistics applied in making the measurement (optional) e.g. average, max, min, mode etc. over the following or preceding interval
Matrix	Indicates the nature of the 'complex' in which the object of interest is held. e.g. biota {Mytilus galloprovincialis (ITIS: 79456: WoRMS 140481) [Subcomponent: flesh]}

Constraint	Some constraints on the situation in which an observation is made (optional) e.g. dry, in-solution.

Table 2 Complex Property facets proposed by Leadbetter and Vodden[1]

These facets and the practices that arise around their use in ‘annotating’ complex properties may be sufficient for UKCEH needs. Alternatively, one could conceive of an open-ended facet space where governed sets of facets and facet values were made available for annotating complex properties. Such annotations make the properties themselves more discoverable and the datasets that use them - provided the usage of properties is catalogued as part of the dataset (or time series metadata).

The INSPIRE’s base model also includes a similarly spirited construct for observable properties which includes facets for unit of measure, base phenomenon, statistical measure and constraints.



Published by European Commission Joint Research Centre [16]

Figure 18 INSPIRE Observable Properties (from INSPIRE Base Models)

Where possible we recommend that feature-of-interest be used primarily to encode the location/place aspects of the observation and that it be orthogonal to the other facets. In some situations, only some aspects of feature location is established by the feature-of-interest. For example, in Climate and Forecasting Standard Names[7], the height which an observation is made may be indicated in the name of the observed property by the embedding of surface qualifiers such as: `at_ground_level, at_maximum_wind_speed_level, at_sea_floor, at_sea_ice_base` (and more). See also the discussion on how to model more specific location on UKCEH sites in section 7.2 Location on site.

Current practices have resulted in the definition of observable properties that have complex meaning. The use of facetted annotations along the lines described by Leadbetter and Vodden[1] lead to the grounding of property definitions in ways that improved their discoverability, their interpretation and their potential for reuse. We advocate the adoption of this approach when defining new observable properties, however this does rely on the establishment of governance around a set of standard facets and the range values that each such facet can have. It is also good practice to create registers of observable properties with the intent of promoting their sharing and reuse in data publications.

5.2 Reference Data Management

Reference data is crucial for the interlinking of datasets (and time series). Typically we think in terms of:

- **Vocabularies:** The standing terminology (classes and properties) used to give expression to such things as observations, sensors, procedures etc.

- **Reference Data:** This covers both codelist/controlled vocabulary terms and collections of data that are of general utility for multiple datasets: for example: Observable Properties; Geographic Features (domain features and sampling features); Units Of Measure; chemical substances and elements; instrument types and instances; procedures and methods.

- **Operational Data:** Observational or Transactional data: Observations and time series data expressed using the vocabulary elements and grounded through the use of reference data.

Shared collections of reference data provide the points of contact between the datasets/time-series that use them in their expression. In the previous section we recommended the use of Leadbetter's complex property model or a generalised variant thereof. However, such an approach relies upon a well curated (and motivated) set of property facets and their values.

In addition to acting as a repository of terms, the NERC Vocabulary Server maintains mappings between synonymous and closely matching terms from other collections, for example Climate and Forecasting standard names[8,9]. These relationships can also serve to ground the meaning of related terms.

Standard names for shared geographical features, such as the nodes and links in a river network and more aggregate constructs such as water bodies, whole rivers[10] and their catchments need to be selected or developed. More ephemeral features - say the cruise track of a research vessel or the track

---

[7] http://cfconventions.org/Data/cf-standard-names/docs/guidelines.html
[8] http://cfconventions.org/Data/cf-standard-names/67/build/cf-standard-name-table.html
[9] http://vocab.nerc.ac.uk/standard_name/
[10] It is somewhat ironic that the OS digital river network (both open and commercial versions) have only nodes and links. There is no aggregated construct for a river. In that sense there are no rivers in the river network, only nodes and links.

of some aerial observation need to be recorded as part of the reference data for the corresponding data sets, but are unlikely to see reuse for future observations - though to within some tolerance and the use of GPS it would be possible to make repeat observations along the same spatial track.

UKCEH should identify core sets of reference data; support their ongoing curation and maintenance; and enhance their discoverability where necessary through the use of semantic 'tagging'..

5.3 Treatment of Sampling

O&M, INSPIRE EMP and SSN/SOSA take different approaches to the treatment of sampling.

O&M introduces the notion of sampling features which can be specialised to include monitoring points and specimens. Sampling features may be related to some domain feature-of-interest via **sampledFeature** and to each other via **relatedSamplingFeature**.

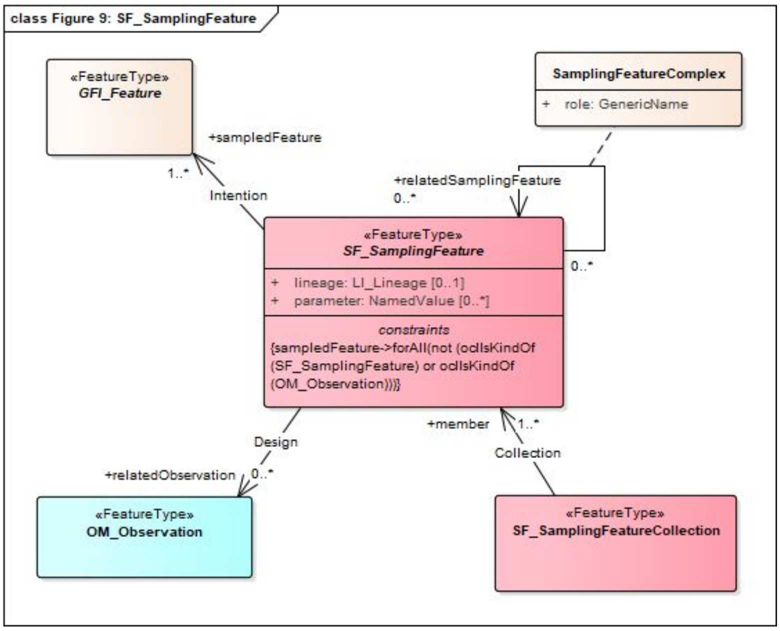
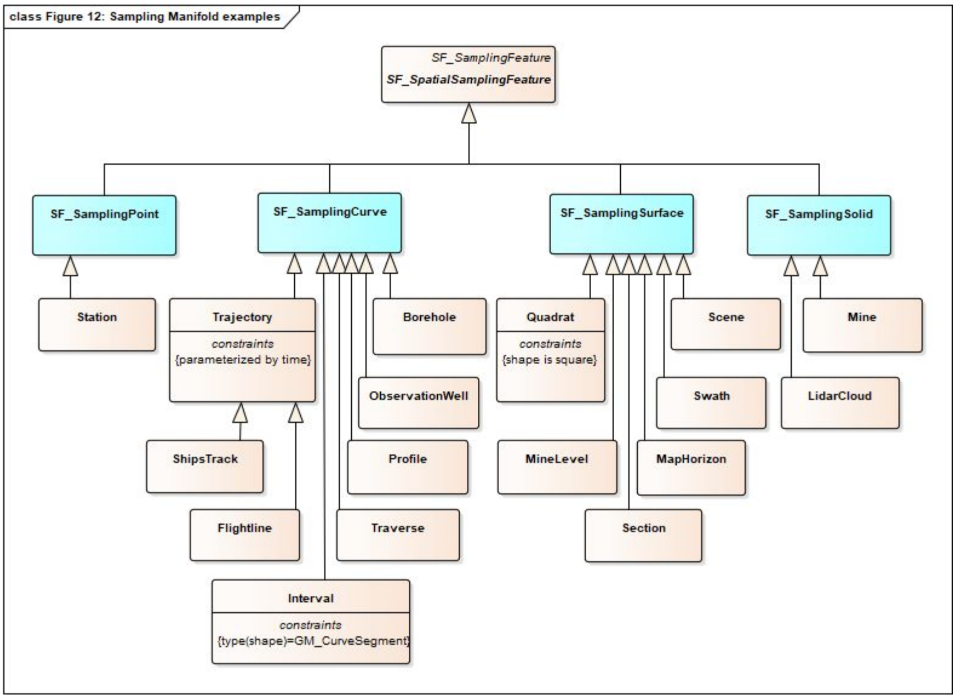


Figure 19 ISO 19156 O&M Figure 9 SF_Sampling Feature

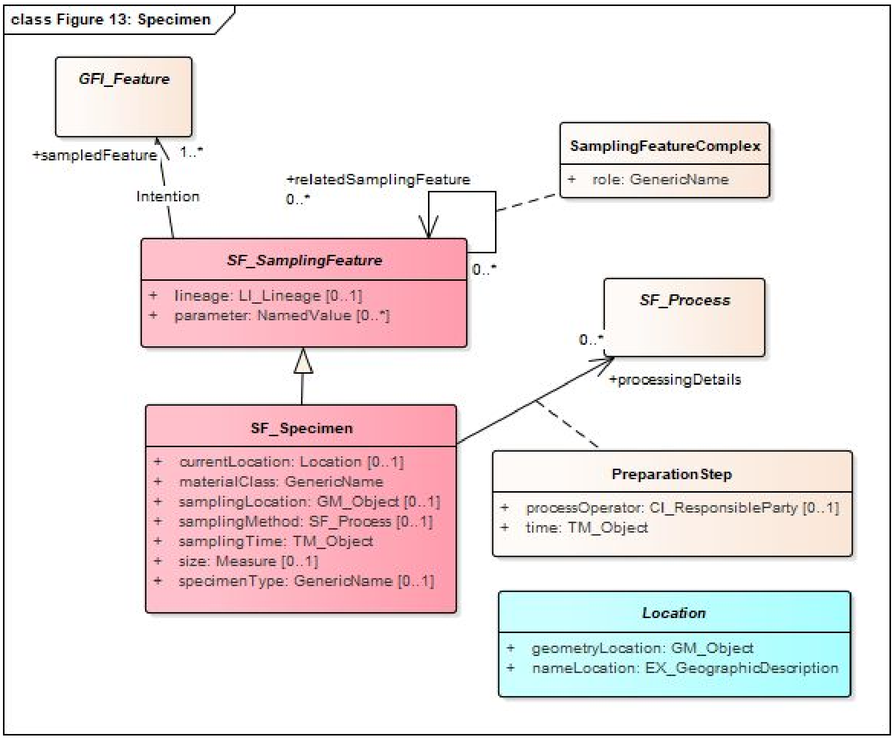
O&M provides some illustrative examples of possible sampling features which themselves represent some level of spatial sampling of a feature (of interest).



[Copyright © 2013 Open Geospatial Consortium](). See [2]

Figure 20 ISO19156 O&M Fig 12 Sampling manifold examples.

O&M also defines **SF_Specimen** to represent physical samples taken from or representative of a feature. Specimens are typically subject to observation ex-situ in a laboratory.



[Copyright © 2013 Open Geospatial Consortium](). See [2]

Figure 21 ISO 19156 O&M Figure 13 Specimen

Neither O&M nor INSPIRE EMF represent the *act* of taking a sample. Instead **SF_Specimen** is used to represent a sample. However, O&M does provide for a specimen to indicate where it was taken (**samplingLocation**), by what method (**samplingMethod**) and when (**samplingTime**). It also provides for the description of steps taken to prepare the sample for observation or measurement (**PreparationStep** and **processingDetails**).

In O&M Specimens may be related to other sampling features such as sampling points and vice versa via **relatedSamplingFeature**. The 'network' of such related sampling features form a sampling feature complex. ISO19156 O&M does not provide clear guidance on whether the ultimate domain feature (of interest) or a relevant sampling feature from the complex (such as a specimen or a monitoring point) should be cited as the feature-of-interest for a given observation. INSPIRE guidance[15] on the use of O&M is more prescriptive and nominates the use of a sampling feature in such cases. In particular, INSPIRE recommends the use of an **SF_Specimen** in the case of ex-situ observation. Related monitoring points and domain features can then be reached via **relatedSamplingFeature** and **sampledFeature** links. In O&M **sampledFeature** is intended to be a direct link to the domain feature-of-interest that is the real target of the observation.

In contrast, the SOS Hydrology profile [21] a recommends instances of **WaterMonitoringPoint** as an observations feature-of-interest.

SSN/SOSA takes a completely different approach as it explicitly models acts of sampling in a similar manner to the way it models acts of observation. The resulting **sosa:Sample** may then be the feature-of-interest for **sosa:Observations** made on the sample. Both the sample's **sosa:isSampleOf** and the **sosa:Sampling**'s **sosa:hasFeatureOfInterest** make the connection to the sampled domain feature. This domain feature may be a monitoring point with further links to an ultimate feature-of-interest for which it stands proxy. Proposed extensions to SSN/SOSA (see 4.2.1 SSN Extensions (SSN-EXT) above) introduce a direct reference to the ultimate feature of interest, **hasUltimateFeatureOfInterest**, which may have much broader scope (UK weather, River Exe, Global Warming) than the proximate feature of interest (a monitoring point, a sampled specimen or a segment of river).

We advocate the explicit representation of sampling introduced by SSN/SOSA. It provides a more direct and coherent representation of the state of affairs in the case when physical sampling occurs. It provides for attriculation of the sampling process and timing separately from that of any measurements observation procedure. See how this could be applied in practice in example 11.7 Water quality data.

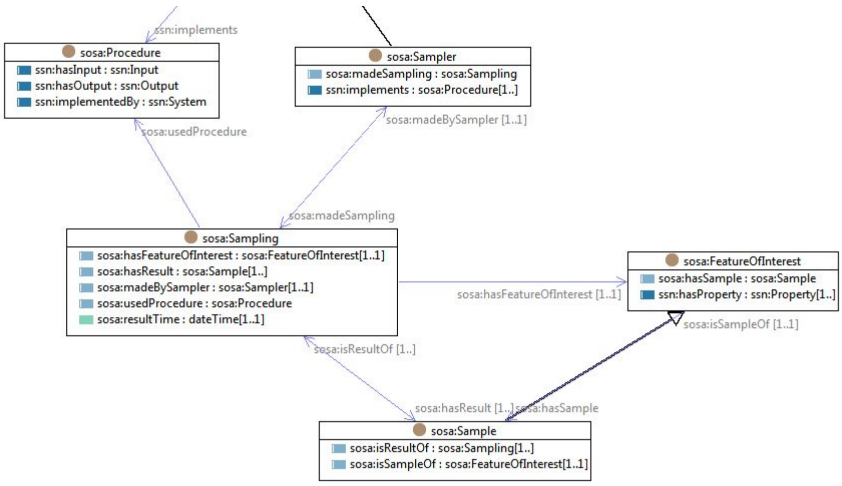


Figure 22 W3C SSN/SOSA Sampling and Sample

5.4 Observations, Time-series, Coverages and Datasets

So far we have largely focussed on individual observations with, mostly, simple largely scalar, results. Here we discuss approaches to forming multiple observations into larger collections.

- Time series are inherently collections of multiple observation results that share a common feature-of-interest and observed properties spanning some possibly open-ended interval of time.

- Dataset is an ill-defined concept that tends to span on-going living datasets (e.g. COSMOS-UK data) as well as more specific point releases (COSMOS-UK data 2018 release), Datasets may contain multiple time series or observation collections and may include copies of associated reference data - referenced features, properties, sensors and sensor metadata.

- Coverages (see section 4.7) are complex structures that can be used to package multiple observation results into a structure that combines spatial and temporal aspects into a single structure. Time series can be implemented as temporal coverages.

5.4.1 Time-series

In practical terms, our notion of a (raw) time series corresponds to a single channel of data reported a data logger. We can define it as a (partial) function of time into the domain of results. Due to the ongoing nature of the environmental monitoring programs in general, and at UKCEH in particular, we frame a time-series as being of indefinite length. Whilst a time-series has a definite past, unless it has concluded, it has an indefinite future. The identity of a time-series should refer to the series over its entire existence. We will refer to shorter extracts of a time-series covering definite time intervals as time-series fragments.

OGC's SOS Hydrology profile[21] recommends the use of **monitoring-point x observed-property x sensor-type** as a key to the identity of an on-going times-series. At UKCEH there can be multiple sensors of the same type (e.g. TDT sensors) at nominally the same monitoring point (e.g. Chimney Meadows - CHIMN). For example COSMOS-UK sites have up to 10 TDT sensors (TDT1-TDT10) each of which measure soil-temperature, relative-permittivity, electrical-conductivity and moisture-content with both raw and cleansed versions of each measurement. Also, over time sensors may be upgraded or replaced (e.g during calibration or maintenance) leading to subtle differences in type. In order to obtain some unique identity for an on-going time series we need to include some expression of finer location within a site (e.g. 15cm depth) or the sensor's role (e.g. for the COSMOS-UK network sensor roles include PA, SNOW_DEPTH, TDT1… TDT10, and more).

In terms of generating some key for identifying a time series there are several time-series characteristics from which it may be constructed:

- ***feature-of-interest*** such as a site or stat*ion*
- ***observed-property*** which may be simple or complex, and if complex it may already incorporate nuisanced facets such as sensor height, depth, statistical and methodological aspect of associated observations.
- ***finer-grained on-site positioning*** with respect to the feature-of-interest (site/station) or ***sensor-role*** (which might also be a reflection of this). For example there may be multiple sensors on a site that observe nominally the same property/phenomenon e.g. soil-temperature, different positions (x,y and z) within a site. Approaches to the expression of fine-grained positioning or sensor-role are explored further in section 7.2 Location on site in the context of some UKCEH datasets.

5.4.2 Datasets

There are multiple conceptualisations of what a dataset is. In academic publication, there is an imperative to enable the reproduction of results as a means to review and verify a conclusion. In this context it is most likely that the content of a dataset is invariant and typically represent a point-in-time snapshot of an otherwise growing dataset. A modern practice is then to assign such snapshots a data DOI (Digital Object Identifier) which serves as a basis for making academic citations of datasets generated or used by a piece of research. UKCEH maintain a number of 'living' datasets. Externally their output is shared as point-in-time dataset releases and as data services accessible via a web based API.

Dataset releases may contain a closure over all the reference data necessary for the interpretation of the main time-series data that they contain. Alternatively, reference data collections, such as for example data about: a river network; a set of hydrological catchments; a collection of monitoring stations in a sensor network etc. can standalone, independent of the observational/time-series data that makes reference to them.

5.4.3 Coverages

Coverages as discussed above conceptually provide a mapping from a spatial/temporal domain to values in the coverage's range. By their very nature, coverages represent a collection of values (results) that are set in a spatial and/or temporal context. Coverage domains are often organised as some form of grid or TIN (triangulated irregular network). The node points of such networks can represent geospatial features with more complex geometries, or there may be a mapping from the local coordinate system of the grid/TIN to a global coordinate system such as WGS 84.

Thus the expansion of the domain of a coverage may necessarily include information about related features, while the expansion of the range of a coverage may necessarily include reference data to support the interpretation of the values in the coverage's range.

As discussed above, Timeseries ML presents a time-series as a single dimensional coverage, with time as that single dimension. Each time-series or time-series fragment forms a collection of observed property values, observed at different times. Such time-series as 'coverage' results may be put together on-the-fly as responses to an API requests.

5.5 Approaches to Process and Sensor Description

OGC O&M does not explicitly cover the description of sensors or their type. Instead it provides for an observation to make reference to the process or procedure that generated the observation. O&M also does not provide the machinery for describing such processes or procedures.

SensorML provides a means to describe composite computational and physical processes (sensors and actuators) and the data flows between them. Sensor ML encourages the use of physical processes to represent individual sensors as the **procedure** referenced from an observation. This results in observation procedure being overloaded with the designation of particular sensor. However, the SOS Hydrology profile moderates that practice and encourages the use of a process representing sensor type (rather than individual sensor).

SSN/SOSA tackles the separation between observation procedure and sensor directly by providing separate attributes for referencing observation procedure (**sosa:usedProcedure**) and the individual sensor (**sosa:madeBySensor**) used in making an observation. Like O&M, SSN/SOSA does not provide the machinery for describing processes and procedures themselves, but it does provide for the description of their inputs and outputs. It also provides structures for describing the operating properties of a system (including sensors) and the conditions in which they hold.

Similarly INSPIRE EMF provides for a **hasObservation** link between an environmental monitoring facility and the observations that it makes. This effectively mirrors the SSN/SOSA connection between an observation and sensor, but in reverse. It also allows multiple facilities (say in a network, station, platform, sensor) hierarchy to reference observations that they have generated.

As described in 4.6 OGC SensorML, SensorML provides an approach for deriving processes from each other, so that a more abstract process may have fewer configuration parameters bound while a more specific process (a particular sensor) has more of its parameters bound and a deployment of a sensor to a particular site or monitoring point has even more of its configuration bound. All this is accomplished with a single **typeOf** relation between process instances. In this way SensorML covers the description of individual sensors, sensor types and their deployments.

Minimally, UKCEH need to be able to capture sensor types and instance with little more than labeling and narrative descriptions. Sensor types will benefit from make/brand and model attributes and sensor instances will need a distinguishing identifier such as a serial number that is expected to be unique relative to sensor type. Both Sensor ML and SSN/SOSA provide for richer descriptions of sensor capabilities and both are capable of more minimal descriptions. Sensor MLs descriptions of sensors as processes spans levels from sensor types, to distinct instances through to time bounded deployments of the same sensor (where associations with features of interest and configuration parameters are resolved for periods of time). SSN/SOSA provides base classes for sensors and deployments. Sensor typing can be accomplished either by specialising **sosa:Sensor** or through the use of coded lists of sensor types that may be held in a register.

There is a certain elegance to Sensor MLs formulation of sensor types, instances and deployments as related physical process. Sensor ML also has more descriptive capability that SSN/SOSA alone. However, if little more is required than distinguishing identifiers and minimal descriptions the more direct use of Sensor, Deployment and, with some invention, SensorType classes would be simpler. Making that choice does not preclude some later enhancement richer Sensor ML descriptions.

5.6 Monitoring Capabilities

INSPIRE EMF, SensorML and SSN/SOSA provide mechanisms that can be used to describe the observing capabilities of monitoring facilities (including sensors). These descriptions can be used for tracking assets; cataloging the properties and features being monitored (as a precursor to discovering observational data sets generated by those facilities); describing the coverage of monitoring programmes.

INSPIRE EMF provides the class **ObservingCapability** for expressing the observing capability of a monitoring network or monitoring facility (e.g. Site/Station, Monitoring Point, Platform, Sensor…) in terms of observable feature, property and procedure combinations monitored by the facility and the time periods for which those capabilities are active (see figure 14 INSPIRE Observing Capability in section 4.5). These feature, property and procedure combinations are are expected to match those used in recording observations made by the facility.

In the case of sensors, **SSN/SOSA** provides attributes to associate a **sosa:Sensor** with observable properties and the observation procedures implemented by the sensor. In SSN/SOSA, observable properties (**sosa:ObservableProperty**) may have a direct connection to a feature-of-interest. However, since the observed property may be a property of many features and we are interested in the feature observed by a sensor, which may change with re-deployments, it is more appropriate that linkage to features-of-interest be established via deployments (**ssn:Deployment**). This would require modest extension of **ssn:Deployment** to reference features-of-interest and the time interval covered by the deployment.

Sensor ML processes (derived from **DescribedObject**) are used for describing individual sensors and sensor types. These processes may carry optional **capabilities, featuresOfInterest** and **validTime** attributes - which in concept cover similar ground to EMF **ObservingCapabilities**. Capabilities are described as SWE data components[11] which themselves may be semantically grounded through their **definition** attribute. Sensor ML uses some combination of free standing events and valid time to represent deployment intervals, however a given process carries only a single set of capabilities, features of interest and valid time. In this context different deployments of the same physical sensor will need to be modelled as different processes. This can be achieved by extending a process chain derivation (**typeOf**) from sensor type, thorough individual sensor instances to their deployments.

[11] SWE data components define the composite shape of data elements input or output by Sensor ML processes and provides a means for their semantic grounding.

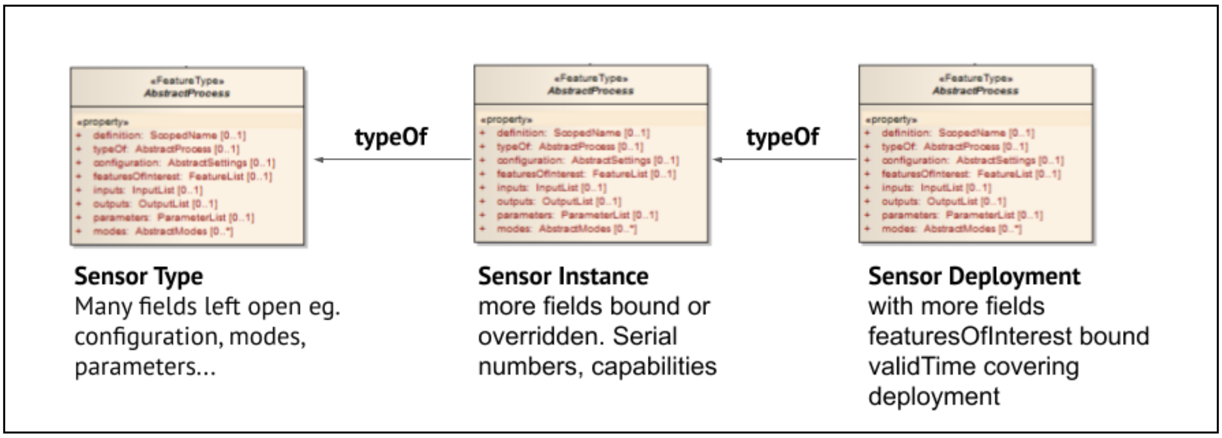


Figure 23 **Sensor Type, Instance and Deployment as Sensor ML Processes.**

INSPIRE EMF, SSN/SOSA and SensorML all provide ways in which the monitoring capability of a sensor (or more generically a facility) can be expressed in terms of feature-of-interest, observed property and procedure used combination. INSPIRE EMF provides the most direct model; SSN/SOSA factors the information across sensors and their deployments; SensorML requires sensor deployments to be framed as processes derived from a physical sensor instance - which feels both 'creative' and somewhat indirect.

5.7 Arrangement of Facilities (Network, Site, Monitoring Point, Sensors, Deployments...)

INSPIRE EMF provides for both **genealogy** and **hierarchy** associations (see figure 13 in section 4.5) for describing the time varying hierarchical arrangement of facilities (Sites, Stations, Monitoring Points, Sensors....) and their succession (facilities being replaced or superseded by other facilities).

SSN/SOSA provides for the hierarchical arrangement of systems and subsystems, similar to INSPIRE EMFs hierarchical arrangement of facilities. However, it only provides for recording the current state. As described in the previous section, SSN/SOSA **ssn:Deployment** requires extension to cover deployment intervals and feature coverage.

Sensor ML provides for the rich composition of aggregate processes from physical, non-physical and aggregate processes and for describing the interconnection of data elements as the inputs and outputs of component processes in such aggregates, see section 4.6. An aggregate can be used to represent, for example, the composition of a COSMOS monitoring site. As **DescribedObjects**, SensorML process have a **validTime** attribute that limits their temporal validity. If, as suggested above, a sensor deployment is modelled as a processes derived from the process representing the physical sensor itself, then it is that deployment process that becomes a component of the aggregate process. As sensors are replaced in an aggregate, the valid time interval of their deployment process is closed, and a new replacement deployment process added to the list of components. In this way, deployment history is retained within the components of the corresponding aggregate process.

INSPIRE EMF provides the most comprehensive model of describing the time varying arrangement of and replacement of facilities. It uses association classes in the UML description for associating time intervals to the broader and narrower relations used to build the hierarchy. **SSN/SOSA** as is adopts a simpler approach using a single **hasSubsystem** property. It would require some extension of the SSN/SOSA model to achieve similar functionality in particular the hierarchical **hasSubsystem** relationship would need to be formulated as an n-ary[12] relationship between a parent, a child and a time interval. It is possible to use **SensorML** in a way records the time varying composition of aggregate processes as described in the previous section.

Where there is a requirement to represent a time-varying arrangement of monitoring facilities then the INSPIRE EMF approach can address this as is. SSN/SOSA would require further extension in order to represent such change over time, however it is suitable for representing a current state model with respect to facilities. For UKCEH the main dynamic aspect appears to be the deployment of sensors to facilities (EMF) or platforms (SSN/SOSA). Again INSPIRE EMF addresses this as is, while SSN/SOSA requires small extension to its Deployment class.

5.8 Data Discovery

It is necessary for users to be able to find data relevant to an enquiry. Their interest may be scoped by some of a number of facets:

- Topic, e.g. what sort of phenomena the dataset covers (e.g. soil moisture and temperature)
- Spatial extent of the dataset: spatial area envelope, or specific geographical features of interests (the River Exe, East Anglia)
- Temporal extent (e.g. two years)
- Spatial and temporal granularity (e.g. 100 sites, 15min profile)
- Provenance and usability information (particular sensor, facility, network or method or created by a particular researcher or project or institution; data quality and usage limitations)

The metadata carried on features, facilities, observations and observation collections (inc. time series) act at a microscopic level. There is a need for these to be summarised to a macroscopic level. The outermost temporal and spatial extent can be modelled as an envelope that covers the relevant spatial extent. If there is a large number of monitored features, the feature content may be covered by cataloging the feature types included or referenced by the data collection.

Leadbetter and Vodden[1] propose **MonitoringProperty** which combines complex property, feature of interest and monitoring procedure, as dataset metadata element. This inherently catalogues the properties, features and procedures associated with a dataset, however the cross product of these facets across a dataset could be very large.

The Complex Property Model (see section 5.1) provides an approach to describing observable properties that makes them discoverable either as a means of promoting reuse across datasets, or as a means to locate observable properties for subsequent use in dataset discovery (find observable properties related to phosphates as a substance of interest and then find datasets that mention any of those properties).

ISO 19115 Metadata [8] provide tools to annotate datasets with
- Types of described features and their properties
- Data quality scope

[12] https://www.w3.org/TR/swbp-n-aryRelations/

- Spatial representation of the dataset
- Identification info (DOIs etc)
- Use limitations
- Application schema information
- Distribution information
- ... and more

As discussed in section 4.2.1, **ObservationCollections** defined in SSN/SOSA extensions enable description of homogeneous sets of observations by instantiating their shared property values. This also means that this data will be available at the level of the collection as a form of summary information. That could include shared topics of the collection (observable property), procedures and even spatial and temporal granularity (e.g. encoded in a complex observable property).

INSPIRE offers **ObservationSet** structure to model groups of observations and their spatial and temporal extent (as defined ISO 19115 Metadata).

Finally, the **PROV** data model and ontology, introduced in section 4.3 can be used to describe the derivation of datasets for one-another, the activities by which they are derived (e.g. INSPIRE EMF environment monitoring activities), and which activities were informed by which datasets etc.

6 UKCEH Datasets

In this section, we'll present models of existing UKCEH data. In section 7, we aim to summarize identified use-cases and demonstrate how they are supported by individual specs. In the next phase of the project, we'll propose a reviewed data model and show how the existing models map to that.

6.1 COSMOS-UK and Greenhouse Gas Network

The current COSMOS-UK / GHG data model approach is to describe data records using predefined data record templates ("Timeseries" in UKCEH terms). These contain explicit or implicit (in description) information about

- Type of object of interest, e.g. soil; air; wind; or even rain gauge (for status)
- Basic phenomenon, e.g. moisture; temperature; direction; status
- Other phenomenon constraints, e.g. type of diagnostic
- Finer location on site, e.g. TDT1; 5cm depth
- Quality information: raw or level2
- Unit of measure, e.g. °C, %
- Statistical aggregation, e.g. instantaneous; 30min average
- Measurement period, e.g. 30 min
- Class of sensors measuring or algorithms computing, e.g. TD; weighted mean method; gamma method

Actual data records then contain

- Reference to site/station
- Value
- Time stamp the value refers to
- And could contain further metadata, e.g. reference to a specific sensor, see 6.2.3

Records are then presented in datasets - at the moment as downloadable spreadsheets.

An overview conceptual diagram is presented in figure 24 below, a complete diagram is available in appendix 12.1 COSMOS-UK Data Model, figure 27. Note that unlike the other two datasets, the COSMOS-UK data model was constructed based on actual data structures currently used (not reverse-engineered from available data), which allows a much finer level of detail.

- thick lines denote main entities and relationships between them.
- colors are used to present related entities
- the entity *Site timeseries* is generated - solid lines denote its membership links (just like solid black lines) and dashed lines indicate which tables it is derived from.

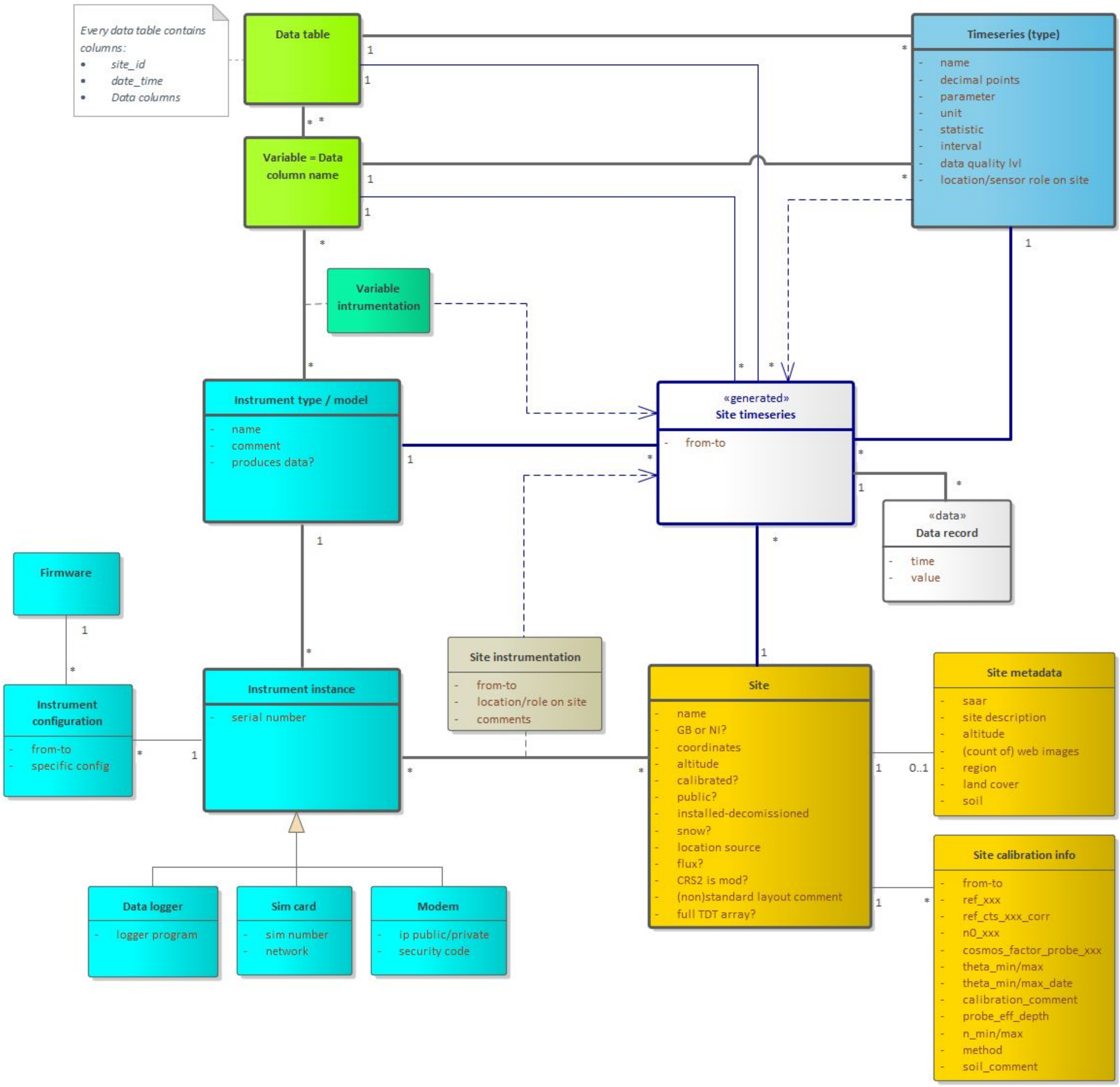


Figure 24 COSMOS high-level data model

6.2 National River Flow Archive

National river flow archive conceptual model, figure 25, was reverse-engineered from https://nrfa.ceh.ac.uk.

The main focus of attention is the description of stations and th#fig_cehnrfadatamodele river catchments they are monitoring. Publically available datasets include

- Archived gauged daily flow data for most stations
- Live checked or unchecked gauged daily flow data managed by the Environment Agency for a few stations
- Peak flow data used e.g. in flood forecasting for some stations

Overview conceptual diagram is presented in figure 25 below, while a complete diagram is available in appendix 12.2 NRFA Data model, figure 28. Colours indicate groups of related information:

- Green, blue = Station and catchment information, respectively
- Turquoise = Archived daily flow data
- Gray = Live (EA) daily flow data
- Red = Peak flow data

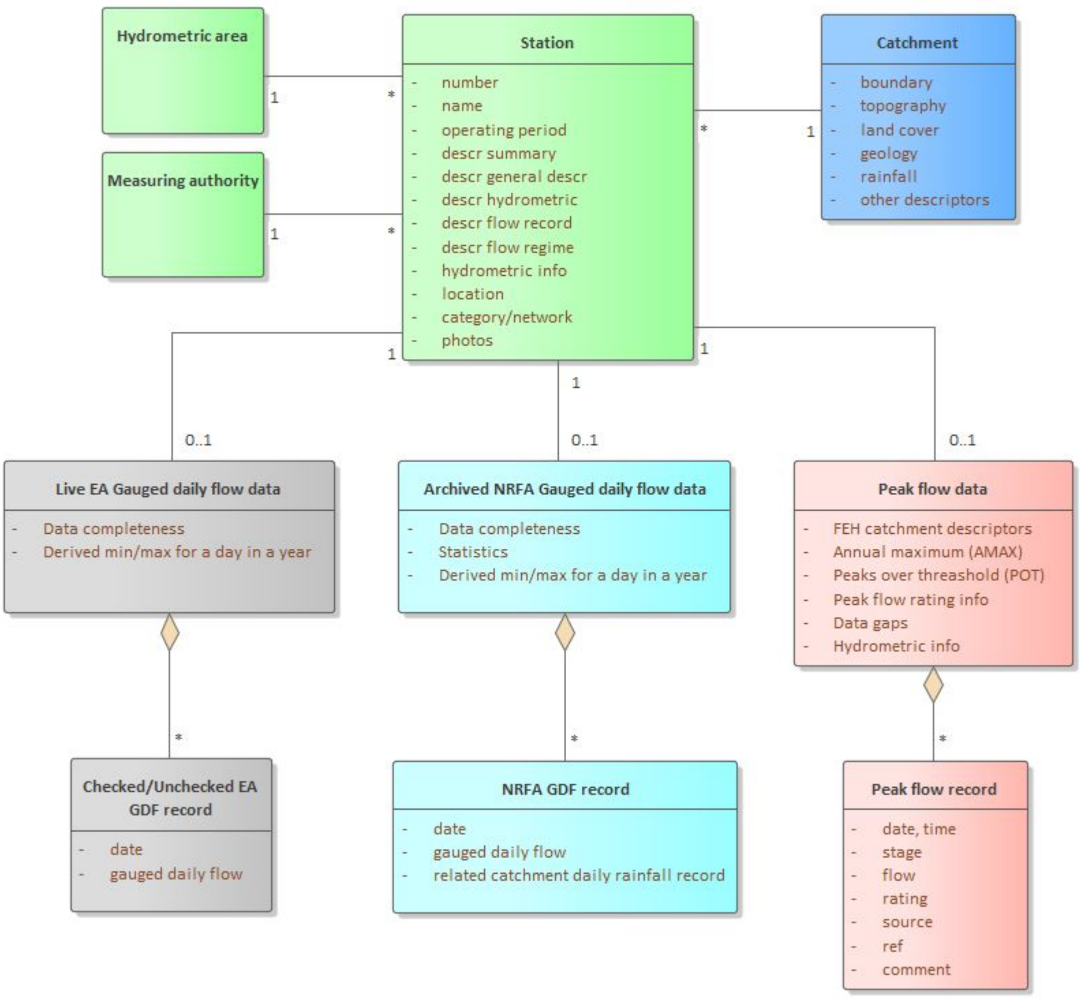


Figure 25 NRFA high-level data model

6.3 Thames Initiative: Water Quality Project

The key aspect of the water quality data is the complex nature of the observed properties, determinands, sampling procedures, sample preparation and analysis steps, all of which form an important part of the domain and significantly affect interpretation of the data.
Here is a high-level conceptual model of the domain, reverse-engineered from the available datasets and use-cases:

- Blue = Rivers and stations
- White = Data records
- Orange = Determinands
- Yellow = Analysis
- Green = Sample preparation

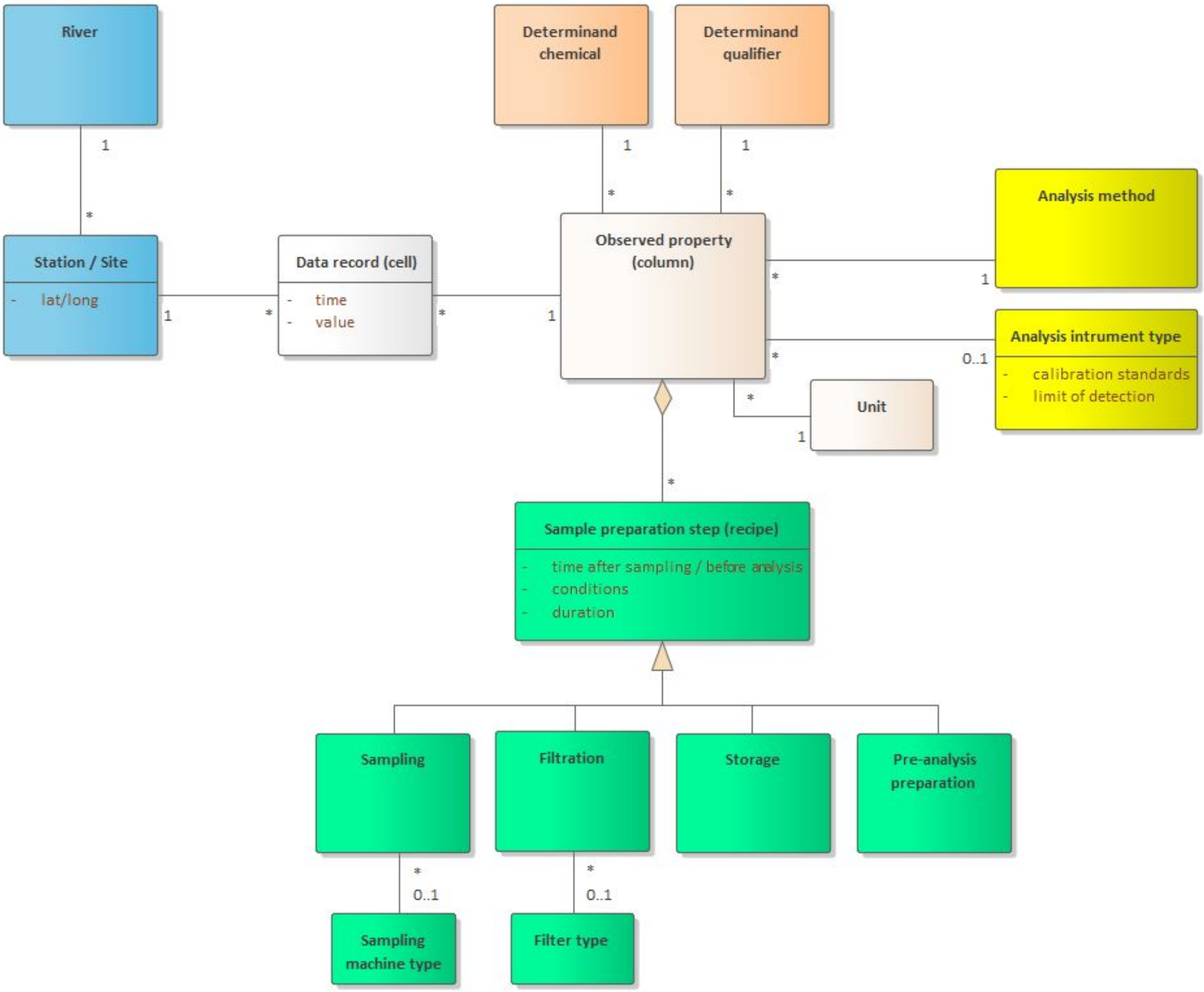


Figure 26 UKCEH Water Quality data model

7 Representing observation metadata

Apart from linking measured values to a specific time, observation (meta)data may indicate:

- Which *geographic* / real-world feature is being monitored
- What particular *property* of it is observed / computed
- At which *site*
- *Where* exactly on the site
- How *frequently*
- *How* it is being monitored
- Instantaneous values, or time *aggregations*
- In what *units* is the value expressed
- What is the *quality* of the value (Which tests has it passed?)

For in-situ sensor measurements, "How" means:

- Using what *sensor*?
  - Model
  - Limit of detection, accuracy
  - Configuration / calibration

For computed data, "How" means:

- From what *data*?
- Using what *algorithm*?

For lab measurements on physical samples (specimens):

- "Property" often needs to include inspected chemical and its form
- "How" covers *sampling*
  - Method
  - From what material
  - When
  - Who
- "How" also covers *lab analysis method* and *machine*
  - Model
  - Limit of detection, accuracy
  - Configuration / calibration

For citizen science, "How?" also covers:

- Who has conducted the measurement / taken the sample?

Most of these won't be explicitly stated on every observation, as discussed in section 7.7 Metadata for larger chunks of data.

7.1 Monitoring sites and geographic features

Features are abstractions of real-world things that can have associated geometries, which describe their shape and placement on the Earth. How do we associate observations with sites / stations / monitoring points and real-world observed features, such as rivers or wetlands? Following the discussion in section 4, let's consider how **sites** can be modelled.

INSPIRE recognizes field stations as Environment Monitoring Facilities (containing individual instruments again modelled as EMF), designed to observe real-world phenomena - this is modelled in their observing capabilities. Facilities can be linked to related observations using a designated **hasObservations** attribute, and the observations linked to the real-world feature being observed via the observation's **featureOfInterest** attribute.

As discussed in section 5.3, in O&M and related specs (especially WaterML, TimeseriesML, SOS Hydro profile but also SOSA/SSN), a site / monitoring point should be modelled as a **SpatialSamplingFeature** (or **sosa:Sample** in case of SOSA/SSN), spatially-sampled from some ultimate feature of interest (e.g. a river) and acting as a proxy for it. There are slight technical nuances in how each spec links proximate and ultimate feature of interest to the observations, see the discussion in section 5.3 and examples in appendices 11.1 and 11.3.

When measurements are performed on surface or groundwater bodies, OGC WaterML2 part 3 [12] and 4 [13], respectively, provide rich vocabulary to model hydrological features and their representations. ELFIE[13] and SELFIE[14] experiments show their use in practice in combination with SOSA/SSN representation of observation data.

7.2 Location on site

Measurements of the same phenomena in different locations / sensor roles on site, e.g. soil moisture at different depths or by different TDT sensors or heat flux plates can be modelled in three ways (in order to distinguish time series, as discussed in section 5.4.1 Time-series):

- Incorporating the distinction in the observed property
- Incorporating the distinction in the feature / monitoring point
- Using a single feature / monitoring point and property, and creating a complex observation result

The first approach requires the creation of distinct properties, e.g. PROFILE_VWC15_RAW and PROFILE_VWC40_RAW to encode the specific location or sensor role. Since these properties have a lot in common, we advocate to use a Complex property model (INSPIRE Observable Properties; or Leadbetter and Vodden[1]) to model the shared (reusable) facets, and express the difference e.g. as a property constraint - see the discussion in section 4.1 and example in appendix 11.1.

The second approach is to create different monitoring points for TDT1 and TDT2 placements at the same site, or similarly for different soil depths. This would foster reusability of properties (generic TSOIL_LEVEL2_MEAN, rather than TDT1_TSOIL_LEVEL2_MEAN) but on the other hand it creates a number of monitoring points (TDT1 at Site_x), see the example in appendix 11.2. These points could then be grouped into collections (SamplingFeatureCollection) representing sites.[15]

[13] https://github.com/opengeospatial/ELFIE
[14] https://github.com/opengeospatial/SELFIE
[15] WaterML states that separate sampling points should only be defined if the particular site is different; where there are multiple observed properties (e.g. multiple sensors being used at one site) should reference the same sampling point. Where sensor locations, i.e. monitoring points are different, we believe this approach can be used.

The third approach requires neither the creation of specific properties nor monitoring sites. It takes the conceptual view where a single (act of) measurement has a complex result, e.g. a coverage for measurements by a temperature probe in different depths. This approach is discussed in section 7.7.2 Complex observation results.

7.3 Sensor data

Sensor capabilities, such as precision or sensitivity, can differ for different sensor models or even configurations. This is a particular issue when interpreting the data affected by service interventions, such as updating a sensor firmware version, recalibration, replacing a malfunctioning sensor or upgrading to a better sensor model. Information about a sensor used to perform individual observations thus represents an important piece of metadata.

UKCEH would like to be able to provide sensor level attribution of at least raw time-series points given that sensors may be replaced during calibration and maintenance activities.

There are at least two ways in which information about sensors placement might be recorded:
- Embedding sensor attribution information into time series data point records
- Keeping separate records of sensor deployment and maintenance

Attribution embedded in data records
Each observation in time-series could be directly associated with the sensor that produced.

```
- "@type":      Observation
  result:        <some value>
  resultTime:   <some date/time>
  observedProperty:    <some property>
  featureOfInterest:   <some feature>
  madeBySensor:        <some sensor>
```

This approach is best aligned with a situation where sensor instance information (e.g. serial number and model) is captured with every time point. However, at UKCEH at least for the time being, information about the recording sensor's identity does not flows alongside sensor readings.

Different specs provide different means to attach sensors to observations. O&M alone suggests that sensors are modelled as observation procedures (class OM_Process). It is however not very prescriptive about whether a process should be a more generic sensor type, or a specific sensor instance.

WaterML Timeseries is similarly non prescriptive, although attributes on WML implementation of OM_Process (ObservationProcess) like gaugeDatum or operatorComments could suggest that the structure is meant to record specific sensor instances. Interestingly, although the Hydro Profile of SOS is intended to standardize use of WaterML in SOS services, as noted above, it explicitly states that a sensor type, *not* instance, should be used as an observation procedure.

INSPIRE EMF (and O&M/SWE guidelines) suggest that sites and sensors should be modelled as Environment Monitoring Facilities and linked to observations using **hasObservation** link. A **Process** class is used to describe a sensor or measurement type. See the example in appendix 11.4 Environment monitoring facilities.

SOSA/SSN has a designated class **Sensor**, which is linked to observations using **madeBySensor** property (see example in appendix 11.3).

SensorML enables modelling of sensor instances, types or even their configurations as processes on various levels of abstraction. These can then either be linked to observations using the O&M procedure link, or their outputs can be wrapped as observation results - see example 11.5 SWE quantities.

W3C PROV data model (and OWL ontology) enables linking of **Entities**, (e.g. Datasets) or **Activities**, (e.g. Observations) to **Agents** (e.g. Sensors) who created / conducted them according to a particular plan (e.g. analytical method or measurement configuration).

Separate deployment records

Records of sensor deployments to monitoring stations are kept that relate a particular sensor with a monitoring station for an interval of time. In serialization, we can

- Completely separate presentation of time-series and sensor deployments - leading to a simpler time-series presentation.

```
- "@type": TimeSeries
  partOf: <indefinite time-series reference> # e.g. CHIMN-TSOIL10-RAW
  featureOfInterest: <some feature>
  observedProperty:  <some property>
  observations:      <array of observations>   # just result value, result timing and qualifiers

- "@type": Deployment
  deployedSystem:     <some sensor>   # Sensor is subclass of System
  deployedOnPlatform: <some platform> # cf EMF facility eg. site/station
  forProperty:        <some observable property>
  deploymentInterval:
  - hasStart:         <some start date or date/time>
    hasEnd:           <some end date or date/time>
```

- Extend the preceding simple presentation with an inventory of sensors used over the time period covered by the serialisation.

```
- "@type": TimeSeries
  partOf: <indefinite time-series reference> # e.g. CHIMN-TSOIL10-RAW
  featureOfInterest: <some feature>
  observedProperty:  <some property>
  observations:      <array of observations>
  sensorsUsed:
  - sensor:        <some sensor>
    interval:
    - hasStart:    <some start date or date/time>
      hasEnd:      <some end date or date/time>
  - sensor:        <some other sensor>
    interval:
    - hasStart:    <some start date or date/time>
      hasEnd:      <some end date or date/time>
```

This approach is more aligned with the way we believe UKCEH currently record sensor deployment information. It makes for more compact storage, since the relation with each observation is not necessarily materialised in the store.

Clearly, it is possible to compute an attribution of time-series readings to originating sensor by aligning monitoring stations and time intervals. However, around the time of a replacement, it is conceivable that some readings may be mis-attributed due to lack of precision in the time recording of such changes.

The ways in which reviewed specs support this include:

- INSPIRE EMF provides “genealogy” links for modelling that a facility has been superseded or replaced by another one. It also enables the hierarchical composition facilities which can be used to associate stations with specific instruments deployments for specific intervals of time. Most importantly, **ObservingCapability** allows efficient description of the time-varying capabilities of a facility to make observations in terms of the properties that can be measured (e.g. soil moisture) with respect to which feature-of-interest (e.g. Site) using a specific process (e.g. measurement configuration or firmware version).

- In SOSA/SSN, Deployment and SystemCapability classes provide similar set of functionalities, however they only record present state and would need to be extended with time scope to model historical records - this is shown in the examples above. Acts of calibration, configuration or other visits could also be recorded as Actuations: changing values of properties of the sensor itself.

- Sensors in SensorML link to currently monitored feature-of-interest and properties (historical monitoring information would require model extensions). SensorML also provides powerful tooling for modelling sensor types or specific sensor configurations as processes on different levels of abstraction, linked together using “typeOf” links. Either of these can then be linked to observations or datasets using the means described above.

- Specific site / station calibrations are also covered in WaterML part 2 [11], in particular provision of rating-conversions: WML suggests that records of historical conversions and related gauging observations are stored as monitoring point metadata.

7.4 Computed data

So far, we have mostly discussed the representation of data directly produced by sensors. In this section, we focus on refined or computed data.

First, we need to express *what* the computed property is. It can be a moderated or validated raw quantity, or have complex meanings, such as the “covariance of properties X and Y”. The complex observable property models described in section 5.1 allows for modelling of such cases, using “constraints” to describe data quality (see the CPM example in appendix 11.1) or aggregations to express combinations of existing properties.

Next, we need to express *How* the data was computed?" This involves modelling processes of quality control, aggregations and the other computations performed on the raw data measured by sensors.

Two specific issues to address are:

- How to model data processing pipelines, e.g. the fact that CTS_MOD_CORR_LEVEL2 data is computed from CTS_MOD_RAW data by Removal of snow days identified from radiation albedo and Gamma correction using site-specific factor.

- How to connect computed data (individual records or whole datasets) to the data it was derived from.

Modelling processing pipelines
An **O&M** procedure can, in the case of computed data, be a reference to a model of a specific algorithm used to create the data. The O&M spec itself leaves the class purely conceptual, various implementations are proposed by other standards.

SOSA/SSN enables modelling of procedure inputs and outputs and also linking to the system implementing the procedure. The model would need extending in order to support the linking of processes into pipelines.

INSPIRE O&M guidelines recommend using either the INSPIRE Generic conceptual model implementation of the Process class, which only provides basic annotation capabilities, or the use of SensorML processes.

SensorML provides a rich tooling for modelling processes, including
- The shape and character of input and output data
- AggregateProcess as a mean to model process pipelines
- Linking of inputs and outputs from different processes
- Modelling of process templates (e.g. interfaces) and their specific implementations (algorithm version 1)

See the example use in appendix 11.6.

Modelling Derivation
The aim here is to capture provenance of the data: which data was used to compute the particular dataset.

O&M has a generic **relatedObservation** attribute to capture that and how two observations are related. The "how" here could be captured using a **computedUsing** attribute.

W3C **PROV** data model (an OWL ontology) addresses specifically this question: What data was used by which agents, participating in what activities and in order to generate some other data. Specifically, the **wasDerivedFrom** link between **Entities** (e.g. datasets or data records), the **wasInformedBy** link between **Activities** (e.g. observations) or the **used** link between an **Activity** (Observation) and **Entity** (Dataset) provide more than sufficient tooling. See the example use in the appendix 11.3.

7.5 Water quality data

Water quality data, in particular ex-situ testing for levels of specific determinands in laboratories, provides the following challenges:

- Modelling the sample and the process of sampling, including people and instruments involved
- Modelling the determinands and laboratory testing processes

Sampling

See discussion on treatment of sampling in section 5.3. In particular, note the O&M sample "preparation steps," including their operators, and SOSA model of sampling, which enables modelling of "Samplers" and procedures.

Determinands

As described in section 5.1, Complex observable properties provides means to express part of the process and metadata information within the property. This is (partly) discouraged by some methodologies, arguing that the details of the observation process should be part of the description of the measurement or sampling process, rather than an observable property.[16] Leadbetter and Vodden's work[1] is building on top of INSPIRE Observable Properties Model, splitting phenomenon into its object of interest (e.g. nitrogen), property (e.g. concentration) and the "matrix" it gets extracted from (e.g. stream sediment soil). Other constraints could include details of process or form of the chemical, (e.g. as Nitrate). See the example in appendix 11.7.

Another specification to mention here is the *O&M and WaterML 2.0 profile for water quality data*[17]*.* The specification itself merely specifies that determinands (observed properties) and units of measure refer to entities defined within the http://environment.data.gov.au/def vocabularies.

7.6 Other metadata

Within the UKCEH data, **data quality** is manifested as:

- On a time series level: raw vs. controlled (level2) time series
- On a data point level: results of quality control tests (e.g. result zero where not permitted)

The former can be incorporated into the observable property, forming a part of a time series identity. Complex property models allow us to specify data quality as a property constraint, see the example in appendix 11.1.

For data point-level quality information, O&M provides a means to express the quality of an observation result as ISO 19115 DataQuality metadata. This enables us to express which evaluation procedures have been applied with which observation results. Since the results of WaterML and TimeseriesML observations are temporal coverages (time series), they provide structures for custom (time) point-level metadata, including qualifiers and quality information (discussed in section 4.4).

---

[16] SensorML: "(...) However, unlike the simple data types in SWE Common, an ObservableProperty does NOT include the properties uom, quality, or constraints, since these are typically characteristics of the measuring procedure and not properties of the observable phenomenon itself."

[17] http://www.opengis.net/doc/BP/watermlwq/1.0

Unit of measure can be recorded with the result (O&M + related, QUDT, SWE); incorporated in a complex observable property; or carried in the observation or time-point metadata (Timeseries-focused specs).

Statistical aggregations (spatial and/or temporal), such as averaging over a specific time interval or expressing a maximum value from a certain area, should again be modelled in complex observable properties, as they form part of a time series identity (D86_1M_WTMN vs. D86_1M_1DAY_WTMN). WaterML and TimeseriesML also provide structures to model aggregations in time series metadata.

7.7 Metadata for larger chunks of data

OGC specs conceptually attach most of the descriptive Information to the act of observation. Due to the amount of repeated information, this may appear impractical. Here we describe some approaches to minimize the repetition of data-descriptive metadata.

7.7.1 Observation collections

Possibly the simplest way to address the problem has just recently been introduced in W3C Extensions to SSN (see section 4.2.1). Following the approach taken in OBOE[18], an **ObservationCollection** class has the ability to carry metadata properties shared by all the homogeneous observations it contains, so that they don't have to be repeated on individual observations. The idea behind is the very same as W3C data cube slices. Collections may be nested, e.g. an outer observation-collection may share an observed-property, procedure and sensor, and contain inner observation-collections at different phenomenon-times, each containing a set of observations on different features-of-interest.
This provides means to model blocks of data with the same metadata (sensor, analysis method; accuracy, ...); series of observations of the same phenomenon on different sites; or complex datasets grouping multiple such series together.

See the example use in appendix 11.3.

7.7.2 Complex observation results

Many UKCEH data time series are closely related. For instance:

- Soil electrical conductivity (or moisture / temperature) at various nominal depths
- Atmospheric pressure: mean and standard deviation
- Wind speed vector
- D86 in various distances
- Data from TDT1-10 sensors

The conceptual idea applicable here is that a single (act of) observation, can have a complex result. This is more likely to be the case for observations made by a single sensor. However with multiple sensors, e.g. TDT sensors or flux plates, we can still model the whole system of sensors as a single complex entity providing the data. There are a couple of advantages to this approach, mainly:

- Related data can be referenced, processed and presented together

[18] https://github.com/NCEAS/oboe

- Data presentation can be more compact

There are several tools available to present complex results:

QUDT QuantityValue provides an easy way to represent measurement uncertainties (stdev).

SWE common data model structures enable the expression and annotation of records, arrays, vectors, matrices or datastreams in various shapes - see the 3D wind direction examples in appendix 11.9, Definitions of data shapes can also be linked to (typically SensorML) definitions of processes as their inputs, outputs or parameters (so we can for instance model shape of results provided by a 3D sonic anemometer).

O&M states that Observation results can be also ISO 19107 **Geometries**: points, curves, surfaces or solids; or **Coverages** - spatial or temporal (time series).

7.7.3 Coverages

Coverages are discussed in section 5.4.3. The approach proposed in O&M and adopted by WaterML Timeseries and TimeseriesML is that an observation result *covers* given points (curves, surfaces, volumes, …) in space and / or time.

An example of a spatial coverage observation might be satellite imaging whose result is e.g. a set of precipitation values for different points in its resolution grid. An example of a temporal coverage observation is any longer-term observation whose result is a bounded time series. An example of a spatio-temporal coverage is the result of a precipitation forecast on 4km grid over the next 2 days.

Coverages could be implemented using pure ISO 19213 Coverage (see section 4.7 and spatial coverage example 11.10); WaterML time series or TimeseriesML (see section 4.4 and example 11.11). Moreover, INSPIRE extensions to O&M also provide specific structures for gridded observations, point observations and trajectory and profile observations, including their time series variants (see example 11.8 Temperature profile). DataCubes are another Coverage-like structure but as discussed in section 3.5, they are much more generic.

Overall, although transferring a bounded time series packaged as an observation result looks tempting, the idea is conceptually wrong *unless there was genuinely a single bounded act of observation* that produced such a result (e.g. a prediction). In particular, as the COSMOS-UK time series are ongoing (note the discussion in section 5.4.1), the observation act has yet to complete. That makes it difficult to say that it has a transferable time-series result.

7.7.4 Avoiding observations

Avoiding the use observations entirely and/or describing data using other means is a viable possibility, although slightly divergent from the main stream of OGC specs and web services like SOS, all built around the Observation concept.

There are multiple approaches to achieve this. SensorML and SWE propose sensor-centric presentation of data: the data are merely treated as sensor output records (or form the content of a sensor output data stream) where sensors carry information about their deployment and measured properties. Data can be enriched with definitions, typically links to an observable property or a monitoring property.

INSPIRE mentions “coverage-centric” approach to data presentation, although it does not give clear guidelines of how such an approach should be used.

8 High Level Recommendations

Our high-level recommendations are centred around the use of concepts drawn from OGC Observations and Measurements (O&M), W3C/OGC SSN/SOSA, INSPIRE Environmental Monitoring Facilities and Sensor ML.

We frame time-series as collections of observations of indefinite length, while concrete serialisations of time-series inevitably convey finite fragment of time-series. Observation collections correspond closely with TimeSeries ML Time Value Pair (TVP) formulation where observations play the role of a TVP. For serialisation purposes we elevate attribute/values that are common to all observations in a collection so that they are carried on the collection rather than on each individual observation. At present we carry timing information for each time-point on the corresponding observation (c.f. TVP), but we are aware that UKCEH have an existing approach to abbreviating result timing information in a way in which it too could be associated with a serialised observation collection as a whole rather than enumerated on each observation.

We anticipate the existence of reference data sets covering geographic features, physical/chemical phenomena, observable properties (complex observable properties and associated controlled vocabularies), units of measure, sensor types, observation procedures and methods, substances and more. Many of these will be interlinked. Collectively they provide the grounding on which to interpret observational data. They also act as points of connection between collections of data. Minimally, reference data may be little more than a unique identifier, a type/classification, a label and a narrative description - however, maximally they can become significant knowledge graphs in their own right. These will require curation and governance by the relevant communities.

The specifications that we have reviewed above overlap in various ways, and share many underlying concepts. The table below indicates the specification source of the main concepts that we draw on.

In the sections that follow we firstly discuss broad approaches to serialisation formats and then illustrate possible serialisations of examples of each of the entity type.

The examples presented below are illustrative and the next stage of the work is expected to develop a more formal model based on these ideas (modulo feedback on this report).

Entity or Concept	Drawn from specification
Observations (individual readings)	A common concept from O&M and SSN/SOSA
Time-Series	Derived from SSN Extensions: **ssn-ext:ObservationCollection**. Using Observation in a similar way to Timeseries ML Time Value Pairs (TVPs). Note that this is slightly different to the TimeseriesML approach, where time series is formulated as a result of a *single* observation.
Observable Properties	Complex Property Model (Leadbetter and Vodden)
Monitoring Facility Descriptions Monitoring Capabilities	INSPIRE EMF and SOSA/SSN: Sites, Stations, Platforms, Monitoring Points...
Sensor Type and Sensor Instance	For minimal descriptions SSN/SOSA **sosa:Sensor** with the addition of curated register of sensor types (reference data). For fuller descriptions Sensor ML use **sml:PhysicalProcess** to represent both sensor types and sensor instances.
Sensor Deployments	SSN/SOSA **ssn:Deployment** extended with time intervals and framed as a subclass of SensorML Events for inclusion in sensor histories.
Samplings and Specimens	SSN/SOSA **sosa:Sampling/sosa:Sample** with explicit representation of the act of sampling, and the resulting sample which itself can be the subject of further observations.
Derivation Process	For simple minimal process descriptions (type, label, description, documentation links) use **sosa:Process** For elaborate composite processes use Sensor ML **sml:AggregateProcess**.

Table 3 Summary Recommendations

8.1 Serialisation Strategy

Information is generally serialised into structures described either as tables, trees or as graphs. Graph serialisations inherently require some form of internal referencing and ideally both graphs and trees will have some form of external referencing to allow references to information held by other data sources, for example shared reference collections such as substances, units of measure, observed properties, features and facilities. There is also a question of whether to refer to simply reference entities, e.g.:

```
- "@type": Observation
  observedProperty: op:TSOIL10-RAW
```

or expand their description in-line, e.g.:

```
- "@type": Observation
  observedProperty:
  # expansion of op:TSOIL10-RAW
  - "@id": op:TSOIL10-RAW
    "@type": ObservableProperty
    label: "Soil Temperature profile, 10cm depth, raw"
    unitOfMeasure:
    - "@id": unit:DegreeCelsius
      "@type": PhysicalUnit
    # ...Nested description of DegreeCelcius
```

Whether to expand references depends on a number of factors:

- How readily available the information required to make the expansion is to the engine generating the serialisation
- Where to halt further expansion.
- Whether there are multiple references to the same entity and whether it is convenient (to the recipient) to repeat the entity multiple times, expand it on first occurrence and make references to it from elsewhere or give no expansion at all.
- Streaming or non-streaming objectives
- Incremental load on the server (both processor and memory)
- Payload size.

A common form of data package, whether from an API or a serialised snapshot of a dataset might be along the following lines:

```
- "@type": Package
  metadata: <information about the response/package itself>
  primaryTopic: <one or more primary topics>
  referenceItems: <referenced items>
```

The package may contain some metadata about itself. For a dataset this may include temporal and spatial envelopes, catalogs of observed features, properties, instruments/sensors used etc.

Primary topics are the main subjects of the response. For a time-series API or dataset they will be time-series, for a features or stations API or dataset they will be features or stations and so-on.

Reference items are the other items that are referenced either from the metadata elements or the primary topics whose expansions are to be included and haven't been elsewhere.

In a degenerate case with a single primary topic all the entity expansion may occur within the tree serialisation rooted at that topic. At the other end of the scale, other than top-level metadata and topic items, all expansions occur in the reference items element. Reference items could have some internal partitioning to separate collections of different kinds of entity (features, observed properties, unit, methods etc.).

However, it is worth repeating that whether, how and where in a serisation process an entity is serialised is often a practical decision about how accessible a description of that entity is to the serialisation engine.

We advocate the use of JSON-LD as a carrier format. It provides an internal and external referencing mechanism, "@id" - so that it can be used to represent both graph and tree structures. It provides mechanisms for grounding field names into URI space which enables an RDF interpretation of the data. In a linked data context we expect the resulting URIs to be dereferencable and to return a description of the referenced entity. This enables reference data and vocabulary information to be curated and published separately from operational data - such as time-series data. Serialisations of operational data may then simply refer to reference items by identifier so that they may be separately retrieved. Alternatively, references items may be expanded in-line within the same serialisation.

In the examples that follow we use YAML notation rather than JSON, partially for compactness and partially because YAML supports embedded comment lines whereas JSON does not. The fragments given can readily by transformed to equivalent JSON using an online tool such as https://www.json2yaml.com

8.2 Time-Series Descriptions

The time-series representations conveyed in API responses or dataset snapshots represent definite fragments of indefinite time-series - indefinite in the sense that they generally extend into the indefinite future. The YAML listing below illustrates a simple serialisation of a time-series fragment, with reference to metadata covering feature-of-interest and observed property.

```yaml
- # A finite serialisation of part of a time series
  "@id": ts:CHIMN-TSOIL10-RAW/R4/2010-01-01T12:00:00Z/PT15M
  "@type": TimeSeries
  partOf: ts:CHIMN-TSOIL10-RAW
  hasFeatureOfInterest: stations:CHIMN
  hasObservedProperty: op:TSOIL10-RAW
  # plus other metadata reference
  hasStart: "2010-01-01T12:00:00Z"
  hasEnd: "2010-01-01T13:00:00Z"
  hasResultPeriod: "PT15M"
  hasObservation:
  - result: 25.0
    resultTime: "2010-01-01T12:15:00Z"
    phenomenonTime: "2010-01-01T12:00:00/PT15M"
  - result: 24.9
    resultTime: "2010-01-01T12:30:00Z"
    phenomenonTime: "2010-01-01T12:15:00/PT15M"
  - result: 24.8
    resultTime: "2010-01-01T12:45:00Z"
    phenomenonTime: "2010-01-01T12:30:00/PT15M"
  - result: 24.7
    resultTime: "2010-01-01T13:00:00Z"
    phenomenonTime: "2010-01-01T12:45:00/PT15M"
```

In this example we illustrate the use of both **resultTime** and **phenomenonTime**, the latter covering the 15 minute interval prior to each result. However, we are aware of UKCEH developed approach using ISO 8601 recurring interval formats to convey timing information in a more compact way, for example from the identifier for the time series fragment the sub-string:

```
R4/2010-01-01T12:00:00Z/PT15M
```

indicates a repetition of four 15 minute intervals starting at noon UTC on 1st January 2010. This could be carried as an explicit attribute of the time series fragment from which result and phenomenon time could be inferred or computed. The specific approach to use will be determined in the next phase of work.

In this example the identifier for the on-going, indefinite, time series, **ts:CHIMN-TSOIL10-RAW**, is derived from the feature-of-interest, **stations:CHIMN**, and observed property, **op:TSOIL10-RAW**, common to all observations in the time series.

In the listing above, references to other entities have not been expanded in-line. In principle they could be expanded - but some consideration needs to be given to how readily accessible the reference information is to the engine assembling the serialisation package and whether it makes the format easier or harder to understand and consume. For example the listing below extends on the listing above by including an expansion of the observed-property, **op:TSOIL10-RAW** discussed in the next section (8.3). The expanded descriptions itself introduces more references to things, e.g. **unit:DegreeCelcius** that could be further expanded in-line - and so-on.

```
- # A finite part of a time series
  "@id": ts:CHIMN-TSOIL10-RAW/R4/2010-01-01T12:00:00Z/PT15M
  "@type": TimeSeries
  # Ongoing indefinite time series
  partOf: ts:CHIMN-TSOIL10-RAW
  hasFeatureOfInterest: stations:CHIMN
  hasObservedProperty:
  - "@id": op:TSOIL10-RAW
    "@type": ObservableProperty
    description:
    - "@value": Soil Temperature profile at 10cm (raw)
      "@language": en
    unitOfMeasure: unit:DegreeCelsius
    objectOfInterest: substances:soil
    property: op:Temperature
    statisticalMeasure:
    - aggregationTimePeriod: periods:PT15M
      aggregation: aggregation:instantaneous
  hasStart: "2010-01-01T12:00:00Z"
  hasEnd: "2010-01-01T13:00:00Z"
  hasResultPeriod: "PT15M"
  hasObservation: <array-of-observations>
```

8.3 Property Descriptions

For property descriptions we recommend the adoption of the complex properties model. This leads to observable properties being annotated with a small number of attributes drawn from controlled vocabularies that match the facets discussed in section 5.1. While property descriptions may be serialised in-line as illustrated above, they are also important reference data and should be published in their own right, much along the lines of the NERC Vocabulary Server. Ideally such property collections serve a significant community and are subject to governance by that community. They, and their meta-properties, serve as a point of connection between data sets and can play a role in their discovery. The YAML listing below illustrates a possible description of the TSOIL10-RAW observable property using properties drawn from the Complex Property Model vocabulary[19]

```
- "@id": op:TSOIL10-RAW
  "@type": ObservableProperty
```

[19] http://purl.org/voc/cpm

```
  description:
  - "@value": Soil Temperature profile at 10cm (raw)
    "@language": en
  unitOfMeasure: unit:DegreeCelsius
  objectOfInterest: substances:soil
  property: op:Temperature
  statisticalMeasure:
  - aggregationTimePeriod: periods:PT15M
    aggregation: aggregation:instantaneous
```

8.4 Feature and Facility Descriptions

For facilities such as monitoring stations and monitoring networks we advocate either the direct use of or specialisation from INSPIRE EMF classes. In particular, monitoring networks such as COSMOS-UK can be modelled as an **emf:EnvironmentalMonitoringNetwork,** for example:

```
- "@id": networks:COSMOS
  "@type": EnvironmentalMonitoringNetwork
  label: COSMOS
  description: "..."
  contains:
  - facility: stations:CHIMN
    network: networks:cosmos
    from:  <from date or date/time>
    until: <until date or date/time>

  - facility: stations:CGARW
    network: networks:cosmos
    from:  <from date or date/time>
    until: <until date or date/time>

    # ... more stations
```

Similarly monitoring stations or monitoring points may be regarded as instances of **emf:EnvironmentalMontoringFacility,** for example the following YAML listing illustrates the partial description of the COSMOS-UK Chimney Meadows station. For illustrative purposes it includes a sensor platform for the stations soil temperature probe and expresses some of its observing capabilities.

```
- "@id": stations:CHIMN
  "@type": EnvironmentalMonitoringFacility
  label: Chimney Meadows COSMOS monitoring station
  specialisedEMFType: emf-type:station #Controlled vocab
  description: "..."

  belongsTo:
  - network: networks:COSMOS
    facility: stations:CHIMN
    from: <from date or date/time>
    until: <until date or date/time> # Omitted whist relation is current.

  representativePoint:
  # these could be collapse to just
  # lat/long/northing/easting attached
  # directly to the station
  - "@type": GM_Point
    crs: epsg:4326  #WGS84
    y: 51.7080 # lat
    x: -1.4788 #long
  - "@type": GM_Point
```

```yaml
    crs: EPSG:27700
    y: 201160 # northing
    x: 436113 #easting

  # EMF time varying hierarchy of subordinate facilities
  narrower:
  - broader: stations:CHIMN  # backlink to parent (optional)
    from: <from date or date/time>
    until: <until date or date/time>  # Omitted whist relation is current.

    facility:
      # Just one facility here - next one carried on next narrower object
      "@id": mp:CHIMN-STP # Soil Temperature Probe
      "@type": EnvironmentalMonitoringFacility
      specialisedEMFType: emf-type:platform
      description: "Chimney Meadows Soil Temperature probe platform"

      observingCapability:
      - from: <from date or date/time>
        until: <until date or date/time>  # Omitted whist relation is current.
        featureOfInterest: stations:CHIMN
        observedProperty: op:TSOIL10-RAW #10cm depth
        procedure: method:COSMOS-STP-PROCEDURE

      - from: <from date or date/time>
        until: <until date or date/time>  # Omitted whist relation is current.
        featureOfInterest: stations:CHIMN
        observedProperty: op:TSOIL20-RAW #20cm depth
        procedure: method:COSMOS-STP-PROCEDURE

      - from: <from date or date/time>
        until: <until date or date/time>  # Omitted whist relation is current.
        featureOfInterest: stations:CHIMN
        observedProperty: op:TSOIL50-RAW #50cm depth
        procedure: method:COSMOS-STP-PROCEDURE
        #... other depths

  - broader: stations:CHIMN  # backlink for next facility...
    #... other more fine grained facilities
```

In the example above the feature-of-interest for the observing capabilities is given as the monitoring-station itself, acting in the role of a sampling feature. However, there may be some other (larger) feature that the monitoring-station serves as a proxy for. This larger feature could be referenced using the **ssn-ext:ultimateFeatureOfInterest** property.

8.5 Sensor and Deployment Descriptions

In an earlier section we advocate the use of SensorML style descriptions for sensor types and sensor instances, and the use of SSN/SOSA style deployment records (extended to include deployment intervals) which can also be framed as Sensor ML events and attached as part of a sensors history. The YAML listing below illustrates sensor type and instance descriptions for a Soil Temperature Profile (STP) sensor, and its deployment at a COSMOS-UK monitoring station.

```yaml
# SensorType (derived from sml:AbstractPhysicalProcess)
- "@id": st:hukseflux-stp-01
  "@type":
  - AbstractPhysicalProcess # SensorML
  - SensorType              # 'local' marker class
  description:
  - "@value":
    "Hukseflux Soil Temperature profile sensor with self-test.
     STP01 accurately measures the temperature profile of the
```

```yaml
      soil at 5 depths close to its surface. It is used for
      scientific grade surface energy balance measurements. The
      sensor is buried and usually cannot be taken to the
      laboratory for calibration. The on-line self-test using
      the incorporated heating wire offers a solution to verify
      STP01's measurement stability."
     "@language": en
  documentation:
  - https://www.hukseflux.com/products/heat-flux-sensors/soil-temperature-sensors/stp01-soil-temperature-sensor
  capabilities:

  # Observable Properties
  - op:TSOIL02-RAW  #  2cm depth
  - op:TSOIL05-RAW  #  5cm depth
  - op:TSOIL10-RAW  # 10cm depth
  - op:TSOIL20-RAW  # 20cm depth
  - op:TSOIL50-RAW  # 50cm depth

# Sensor instance (derived from sosa:Sensor and sml:PhysicalProcess)
- "@id": hf-stp:123456
  "@type":
  - PhysicalProcess    # SensorML
  - Sensor        # SSN/SOSA
  typeOf: st:hukseflux-stp-01
  description: "Hukseflux STP 01 sensor - serial no: 123456"
  characteristics:
  # SWE style serial number
  - "@type": Text
    definition: char:serialNo
    label: Serial Number
    value: 123456
  # Simple direct encoding of a serial number.
  - serialNumber: 123456
  - history:
    # List of events including calibrations, deployments...
    - deployments:hf-stp-123456-2013-10-01

# Deployment event (derived from ssn:Deployment and sml:Event)
- "@id": deployments:hf-stp-123456-2013-10-01
  "@type":
  - Deployment        # SSN/SOSA
  - Event        # Sensor ML
  deployedOnPlatform: mp:CHIMN-STP
  deployedSystem: hf-stp:123456
  forProperty:
  - op:TSOIL02-RAW  #  2cm depth
  - op:TSOIL05-RAW  #  5cm depth
  - op:TSOIL10-RAW  # 10cm depth
  - op:TSOIL20-RAW  # 20cm depth
  - op:TSOIL50-RAW  # 50cm depth
  deploymentInterval:
  - hasStart: 2013-10-01
    # hasEnd: <tdb>  #
```

Whilst we advocate the use of deployment records as a means to determine the sensor origin of individual readings, we illustrate three alternatives for incorporating sensor origin in-line in a time-series based serialisation - firstly by composing the serialised time-series from multiple sub-series each of which references a different made-by sensor.

```yaml
- "@id": ts:CHIMN-TSOIL10-RAW/R4/2010-01-01T12:00:00Z
  "@type": TimeSeries
  observedProperty: op:TSOIL10-RAW
  hasFeatureOfInterest: station:CHIMN
  # Ongoing indefinite time series
  partOf: ts:CHIMN-TSOIL10-RAW
  hasMember: # from ssn-ext:
```

```yaml
# 1st definite time series (2 readings from 12:00)
- "@id": ts:CHIMN-TSOIL10-RAW/R2/2010-01-01T12:00:00Z
  "@type": TimeSeries
  madeBySensor: hf-stp:123456
  hasObservation:
  - result: "25.0"
    resultTime: "2010-01-01T12:15:00Z"
    phenomenonTime: "2010-01-01T:12:00:00Z/PT15M"
  - result: "24.9"
    resultTime: "2010-01-01T12:30:00Z"
    phenomenonTime: "2010-01-01T:12:14:00Z/PT15M"

# 2nd definite time series (2 readings from 12:30)
- "@id": ts:CHIMN-TSOIL10-RAW/R2/2010-01-01T12:30:00Z
  "@type": TimeSeries
  madeBySensor: hf-stp:234567
  hasObservation:
  - result: "24.8"
    resultTime: "2010-01-01T12:45:00Z"
    phenomenonTime: "2010-01-01T:12:30:00Z/PT15M"
  - result: "24.7"
    resultTime: "2010-01-01T13:00:00Z"
    phenomenonTime: "2010-01-01T:12:45:00Z/PT15M"
```

Secondly where each observation carries its own direct reference to its sensor of origin:

```yaml
- "@id": ts:CHIMN-TSOIL10-RAW/R4/2010-01-01T12:00:00Z
  "@type": TimeSeries
  observedProperty: op:TSOIL10-RAW
  hasFeatureOfInterest: station:CHIMN
  # Ongoing indefinite time series
  partOf: ts:CHIMN-TSOIL10-RAW
  hasObservation:
  - result: "25.0"
    resultTime: "2010-01-01T12:15:00Z"
    phenomenonTime: "2010-01-01T:12:00:00Z/PT15M"
    madeBySensor: hf-stp:123456
  - result: "24.9"
    resultTime: "2010-01-01T12:30:00Z"
    phenomenonTime: "2010-01-01T:12:15:00Z/PT15M"
    madeBySensor: hf-stp:123456
  - result: "24.7"
    resultTime: "2010-01-01T12:45:00Z"
    phenomenonTime: "2010-01-01T:12:30:00Z/PT15M"
    madeBySensor: hf-stp:234567
  - result: "24.5"
    resultTime: "2010-01-01T13:00:00Z"
    phenomenonTime: "2010-01-01T:12:45:00Z/PT15M"
    madeBySensor: hf-stp:234567
```

Lastly, where an inventory of the relevant sensors used in making the observations is included as an attribute of the time series fragment being transferred,

```yaml
- “@id”: ts:CHIMN-TSOIL10-RAW/R4/2010-01-01T12:00:00Z
  "@type": TimeSeries
  featureOfInterest: station:CHIMN
  observedProperty: op:TSOIL10-RAW
  # Ongoing indefinite time series
  partOf: ts:CHIMN-TSOIL10-RAW

  sensorsUsed
  - sensor: hf-stp:123456
    interval:
    - #hasStart: <some start date or date/time>
      hasEnd: "2010-01-01T12:30:00Z"
  - sensor: hf-stp:234567
    interval:
```

```
    - hasStart: "2010-01-01T12:30:00Z"
      #hasEnd: <some end date or date/time>

  hasObservation:
  - result: "25.0"
    resultTime: "2010-01-01T12:15:00Z"
    phenomenonTime: "2010-01-01T:12:00:00Z/PT15M"
  - result: "24.9"
    resultTime: "2010-01-01T12:30:00Z"
    phenomenonTime: "2010-01-01T:12:15:00Z/PT15M"
  - result: "24.7"
    resultTime: "2010-01-01T12:45:00Z"
    phenomenonTime: "2010-01-01T:12:30:00Z/PT15M"
  - result: "24.5"
    resultTime: "2010-01-01T13:00:00Z"
    phenomenonTime: "2010-01-01T:12:45:00Z/PT15M"
```

9 Next Steps

1. **Propose detailed data model.** Based on the high-level recommendations outlined in the previous section, a detailed model will be created and its alignment with reviewed standards documented and formally modelled. We'd advocate that the model is created as an OWL ontology. In order to do that, we need to:
    a. Outline intended model structure and definitions. This can be done in the form of an annotated UML diagram.
    b. Decide which existing vocabularies to use and propose alignment. That is - which classes and properties are going to be equivalent to or subclasses / subproperties of which structures introduced in the reviewed specs
    c. Propose URIs for new vocabulary entities
    d. Create the OWL ontology.
2. **Resolve reference data management.** That means deciding on what external reference data is going to be used and what will have to be created and managed by the UKCEH. This could involve broader discussion between the curators of the existing reference data vocabularies.
3. **Benchmark on UKCEH (meta)data.** This means modelling metadata for different UKCEH use cases as proposed and linking them to existing concepts in external vocabularies. This should help reveal potential weaknesses of the approach and help to form enhanced iterations of the model.
4. **Propose data model serialization and transfer syntax.**
5. **Document** the model, examples, data discovery and access patterns - describe use cases such as finding datasets about nitrogen concentration in Welsh rivers, including the format of the responses.

10 References

[1] Leadbetter, Adam M., and Peter N. Vodden. "Semantic linking of complex properties, monitoring processes and facilities in web-based representations of the environment." International Journal of Digital Earth 9.3 (2016): 300-324, DOI: [10.1080/17538947.2015.1033483](10.1080/17538947.2015.1033483)

[2] Cox, Simon. "Geographic Information: Observations and Measurements–OGC Abstract Specification Topic 20." (2010). http://portal.opengeospatial.org/files/?artifact_id=41579

[3] Cox, Simon. "Observations and Measurements-XML Implementation. Version 2.0." (2011). http://portal.opengeospatial.org/files/?artifact_id=41510

[4] Tomkins, James, and Dominic Lowe. "Timeseries Profile of Observations and Measurements, Version 1.0." (2016). http://docs.opengeospatial.org/is/15-043r3/15-043r3.html
[5] Haller, A., et al. "Semantic Sensor Network Ontology, W3C Recommendation 19 October 2017." https://www.w3.org/TR/vocab-ssn/
[6] Cox, Simon. "Extensions to the Semantic Sensor Network Ontology, W3C First Public Working Draft 22 November 2018." https://www.w3.org/TR/vocab-ssn-ext/
[7] Lebo, Timothy, et al. "Prov-o: The prov ontology." W3C recommendation (2013). http://www.w3.org/TR/prov-o/
[8] ISO/TC 211. "ISO 19115-1: 2014. Geographic Information–Metadata–Part 1: Fundamentals." The International Organization for Standardization: Geneva, Switzerland (2014).
[9] Tomkins, James, and Dominic Lowe. "Timeseries Profile of Observations and Measurements, Version 1.0." (2016). http://docs.opengeospatial.org/is/15-043r3/15-043r3.html
[10] Taylor, Peter. "OGC WaterML 2.0: Part 1-Timeseries." (2012). https://portal.opengeospatial.org/files/?artifact_id=57222
[11] Taylor, Peter. "OGC WaterML2. 0: Part 2-Ratings, Gaugings and Sections, Version 1.0." (2016). http://docs.opengeospatial.org/is/15-018r2/15-018r2.html
[12] Blodgett, David, and Irina Dornblut. "OGC WaterML 2: Part 3-Surface Hydrology Features (HY_Features). Version 1.0." (2018). https://portal.opengeospatial.org/files/?artifact_id=57222
[13] Brodaric, Boyan. "OGC WaterML 2: Part 4–GroundWaterML 2 (GWML2), Version 2.2." (2017). https://portal.opengeospatial.org/files/?artifact_id=57222
[14] INSPIRE Thematic Working Group Environmental Monitoring Facilities. "D2.8.II/III.7 INSPIRE Data Specification on Environmental Monitoring Facilities – Technical Guidelines." (2013). https://inspire.ec.europa.eu/id/document/tg/ef
[15] INSPIRE Maintenance and Implementation Group (MIG). "D2.9 Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE." (2016). https://inspire.ec.europa.eu/id/document/tg/d2.9-o%26m-swe
[16] INSPIRE Maintenance and Implementation Group (MIG). "INSPIRE data models", Web Site, https://inspire.ec.europa.eu/Data-Models/Data-Specifications/2892
[17] Botts, Mike, et al. "OGC sensorML: model and XML encoding standard." Technical Standard 2.12-000 (2014). https://portal.opengeospatial.org/files/?artifact_id=55939
[18] Robin, Alexandre. "OGC SWE Common Data Model Encoding Standard. Version 2.0.0." (2011). https://www.opengeospatial.org/standards/swecommon
[19] ISO/TC 2011. "ISO 19123: Geographic information—Schema for coverage geometry and functions." The International Organization for Standardization: Geneva, Switzerland (2005). https://portal.opengeospatial.org/files/?artifact_id=19820
[20] Cyganiak, Richard, and Reynolds, Dave. "The RDF Data Cube Vocabulary W3C Recommendation 16 January 2014." https://www.w3.org/TR/2014/REC-vocab-data-cube-20140116/
[21] Andres, Volker, Simon Jirka, and Michael Utech. "OGC Sensor Observation Service 2.0 Hydrology Profile. Version 1.0." (2014). http://docs.opengeospatial.org/bp/14-004r1/14-004r1.html

11 Appendix A Examples

In this annex, we provide a set of illustrative domain examples modelled using approaches described in reviewed specs. Examples are encoded in json-ld-like yaml, to be able to include comments in the code, make examples more compact and readable. Each example can be easily converted to json using some format conversion tool, e.g. https://json2yaml.com.

OGC specs in particular are oriented around XML encoding and the translation to json / yaml is not always unambiguous. Hence, the examples should be treated as merely illustrative of how different situations can be modelled and elements linked, rather than as a normative encoding format. Prefixes are mostly used to indicate which model each entity / relationship comes from.

11.1 O&M observation: Location on site in complex observable property

```yaml
# This resonates with the current UKCEH approach.
# It is simple but requires representing more information in the observed property (depth = 10cm)

# Modelled according to Adam Leadbetter's complex property model (CPM) specification,
# very similar for INSPIRE Observable Property Model

---
# property defined as one of the existing UKCEH time series
- "@id": property:STP_TSOIL10_RAW
  "@type": cpm:ObservableProperty
  rdfs:label: Soil temperature profile (10 cm depth) (raw)
  cpm:objectOfInterest: sweet:realmSoil
  cpm:property: sweet:propTemperature
  cpm:uom: unit:cm
  cpm:statisticalMeasure:
    cpm:aggregation: func:avg
    cpm:aggregationTimePeriod: interval:30min
  cpm:constraint:
  - cpm:constraintProperty: property:depth_cm
    cpm:value: 10
  - cpm:constraintProperty: dataQuality
    cpm:value: raw

# site
- "@type": sf:SpatialSamplingFeature
  "@id": site:CHIMN
  sf:sampledFeature: region:SouthEastEngland    # could also be a specific river
  sf:hostedProcedure:
      - sensor:Hukseflux_STP01_5941
    # ...
  gm:geometry:
    "@type": gm:Polygon

# observation
- "@type": om:Measurement
  om:featureOfInterest: site:CHIMN
  om:observedProperty: property:STP_TSOIL10_RAW
  om:procedure: sensor:Hukseflux_STP01_5941
  om:phenomenonTime: 2019-09-01T00:00:00Z
  om:result: 15.6
```

11.2 O&M observation: Location on site in feature-of-interest

```yaml
# We are ultimately looking at soil (or UK soil) properties,
# like moisture, temperature, etc. which are observed at some
# sampling points - in this version the sample points are specific locations
# within a site where sensors are located.

# This is a useful way to keep properties generic and more reusable:
# The approach could be used to solve e.g. TDT1-10 data distinction
# or measurements in different depths - this example

---
# site
- "@type": sf:SpatialSamplingFeature
  "@id": site:CHIMN
  sf:sampledFeature: region:SouthEastEngland    # could also be a specific river
  sf:hostedProcedure:
      - sensor:Hukseflux_STP01_5941
    # ...
  gm:geometry:
    "@type": gm:Polygon

# soil at 10cm depth on CHIMN
- "@type": sf:SpatialSamplingFeature
  "@id": site_loc:soil_at_10cm_depth_CHIMN
  sf:sampledFeature: sweet:realmSoil
  sf:relatedSamplingFeature: site:CHIMN     # “sampled” site here, as discussed in section 4.3 Treatment of Sampling
  sf:shape: depth:10cm

# observation
- "@type": om:Measurement
  om:featureOfInterest: site_loc:soil_at_10cm_depth_CHIMN
  om:observedProperty: property:STP_TSOIL_RAW
  om:procedure: sensor:Hukseflux_STP01_5941
  om:phenomenonTime: 2019-09-01T00:00:00Z
  om:result: 15.6
```

11.3 SOSA/SSN observations - raw and level2

```yaml
# In this example, we'll demonstrate the use of SOSA/SSN,
# instead of O&M but an alignment module between the two exists.

# We'll show two datasets, modelled as nested observation collections,
# which are used to carry property values shared by all observations contained.

---
# This dataset contains raw sensor observations
- "@id": dataset:bf5435ea-7387-4a03-9c7e-b91dea1236d4
  "@type": ssn-ext:ObservationCollection
  sosa:observedProperty: observableProperty:STP_TSOIL10_RAW
  sosa:usedProcedure: sensorType:Hukseflux_STP01

  # Useful relation introduced in SSN extensions - this could also point to a river
  ssn-ext:hasUltimateFeatureOfInterest: region:SouthEastEngland

  qudt:unit: unit:DEG_C
  ssn-ext:hasMember:
  - "@type": ssn-ext:ObservationCollection
    sosa:hasFeatureOfInterest: site:CHIMN
    sosa:madeBySensor: sensor:Hukseflux_STP01_5941
    ssn-ext:hasMember:
    - "@type": sosa:Observation
      sosa:hasResult: '19.0'
      sosa:resultTime: '2019-09-01T00:00:00Z'
```

```yaml
    - "@type": sosa:Observation
      sosa:hasResult: '18.0'
      sosa:resultTime: '2019-09-01T00:30:00Z'
    # ...


# This dataset contains level2 observations - data that passed quality checks
- "@id": dataset:1eb5a86a-8cb6-4175-887a-cc1cf66b326d
  "@type": ssn-ext:ObservationCollection

  # Dataset was derived from the "raw" one
  prov:wasDerivedFrom: dataset:bf5435ea-7387-4a03-9c7e-b91dea1236d4

  # Quality control procedure can be modelled e.g. as SensorML process
  sosa:usedProcedure: algorithm:QualityControl_STP_v1

  sosa:observedProperty: observableProperty:STP_TSOIL10_LEVEL2
  ssn-ext:hasUltimateFeatureOfInterest: region:SouthEastEngland
  qudt:unit: unit:DEG_C
  ssn-ext:hasMember:
  - "@type": ssn-ext:ObservationCollection
    sosa:hasFeatureOfInterest: site:CHIMN
    ssn-ext:hasMember:
    - "@type": sosa:Observation
      sosa:hasResult: '19.0'
      sosa:resultTime: '2019-09-01T00:00:00Z'
    - "@type": sosa:Observation
      sosa:hasResult: '18.0'
      sosa:resultTime: '2019-09-01T00:30:00Z'
    # ...
```

11.4 Environment monitoring facilities

```yaml
# Station and sensors installed there are modelled as facilities.
# Types (models) of sensors are modelled as procedures

- "@id": station:CHIMN
  "@type": emf:EnvironmentMonitoringFacility
  emf:specialisedType: station
  emf:observingCapability:
    - "@type": emf:Capability
      emf:featureOfInterest: site:CHIMN
      emf:observedProperty: property:TDT1_VWC_RAW
      emf:procedure: model:Acclima_Digital_TDT
  # - ...
  emf:narrower:
    - "@id": sensor:Acclima_Digital_TDT_4589
      "@type": emf:EnvironmentMonitoringFacility
      emf:specialisedType: sensor
      model: model:Acclima_Digital_TDT

      # Note that the observations are connected to EMF using EMF "hasObservation" link
      # rather than using the OM "procedure" link. In EMF, procedures are meant to remain generic
      emf:hasObservation:
        - "@type": om:Measurement
          # ...
          om:procedure : model:Acclima_Digital_TDT
          # ...

      # - …
```

11.5 SWE quantities

```yaml
# Using SWE common data model quantities.

# Observable Property. Note that according to SWE guidelines, this doesn't contain details of the
# observation process: specific units, quality or format of nil values are described in the relevant quantity
# produced by the sensor
- "@id": property:STP_TSOIL10
  "@type": cpm:ObservableProperty
  rdfs:label: Soil temperature profile (10 cm depth)
  cpm:objectOfInterest: sweet:realmSoil
  cpm:property: sweet:propTemperature
  cpm:statisticalMeasure:
    cpm:aggregation: func:avg
    cpm:aggregationTimePeriod: interval:30min
  cpm:constraint:
  - cpm:constraintProperty: property:depth_cm
    cpm:value: 10

# Quantity (soil temperature) produced by soil temp probe. Includes details of format, units and quality.
- "@id": quantity:STP_TSOIL10_RAW
  "@type": swe:Quantity
  rdfs:label: Raw soil temperature profile (10 cm depth) from STP
  swe:definition: property:STP_TSOIL10
  swe:quality: quality:raw
  swe:uom: unit:degC
  swe:nilValues: n/a
  # precision, sensitivity and more could be included here

# Specific sensor modelled as a SensorML process. Format of inputs and outputs could also be recorded on
# the model of sensor type, to which specific sensors can be connected using typeOf link.
- "@id": sensor:Hukseflux_STP01_5941
  "@type": sml:PhysicalComponent
  sml:input:
    # According to SensorML spec, observable properties should be modelled as inputs of sensors
    - property:STP_TSOIL10_RAW
  sml:output:
    # A way of attaching sensors to measurements is defining their results as sensor outputs
    - result:9c960b51-a454-42de-8673-ce45b66b48d9
  # - ...


# Measurement, whose result is formulated as SWE common data model quantity
- "@type": om:Measurement
  # ...
  om:observedProperty: property:STP_TSOIL10     # implicit from the sensor model
  om:result:
    "@id":  result:9c960b51-a454-42de-8673-ce45b66b48d9
    "@type": swe:Quantity                       # implicit from the sensor model
    swe:definition: quantity:STP_TSOIL10_RAW    # implicit from the sensor model
    swe:value: 15.3
  # …
```

11.6 Computation process

Compact:

```yaml
# This example demonstrates modelling of a complex computation process
# "Computation of Volumetric Water Content from raw neutron counts" using SensorML.
# We're using SWE quantities to represent data shape - those can be defined using observable properties

---
"@id": algorithm:vwc_from_cts_raw
"@type": sml:AggregateProcess
rdfs:label: Computation of Volumetric Water Content from raw neutron counts
sml:input: quantity:BACK_CTS_RAW
sml:output: quantity:LEVEL2_VWC_GAMMA_1DAY
sml:components:

# First, we need to get moderated and corrected neutron counts
- "@type": sml:AggregateProcess
  sml:input: quantity:BACK_CTS_RAW
  sml:output: quantity:CTS_MOD_CORR_LEVEL2
  sml:components:

  # To do that, we first need to compute moderated neutron counts
  - "@type": sml:AggregateProcess
    sml:input: quantity:BACK_CTS_RAW
    sml:output: quantity:CTS_MOD_RAW
    sml:components:

    # We do do by removing background variability ...
    - "@type": sml:SimpleProcess
      rdfs:label: Removal of background variability

    # ... and atmospheric effects, for which we need relevant series of data
    - "@type": sml:SimpleProcess
      rdfs:label: Removal of atmospheric effects
      sml:input:
      - quantity:BACK_CTS_RAW
      - quantity:PA_LEVEL2
      - quantity:RH_LEVEL2
      sml:output: quantity:CTS_MOD_RAW

  # With the moderated counts, we need to remove snow days ...
  - "@type": sml:SimpleProcess
    rdfs:label: Removal of snow days identified from radiation albedo

  # ... and apply gamma correction in order to get moderated and corrected neutron counts
  - "@type": sml:SimpleProcess
    rdfs:label: Gamma correction using site-specific factor
    sml:input: quantity:CTS_MOD_RAW
    sml:output: quantity:CTS_MOD_CORR_LEVEL2

# Finally, we'll compute volumetric water content from moderated and corrected neutron counts
- "@type": sml:SimpleProcess
  rdfs:label: Neutron count to VWC conversion using site-specific calibration data
  sml:input: quantity:CTS_MOD_CORR_LEVEL2
  sml:output: quantity:LEVEL2_VWC_GAMMA_1DAY
```

Same example - more verbose, including connections between inputs and outputs:

```yaml
---
"@id": algorithm:vwc_from_cts_raw
"@type": sml:AggregateProcess
sml:name: vwc_from_cts_raw
rdfs:label: Computation of Volumetric Water Content from raw neutron counts
sml:input:
```

```yaml
- sml:name: cts
  swe:quantity: quantity:BACK_CTS_RAW
sml:output:
- sml:name: vwc
  swe:quantity: quantity:LEVEL2_VWC_GAMMA_1DAY
sml:components:
- "@type": sml:AggregateProcess
  sml:name: cts_correction
  sml:input:
  - sml:name: cts
    swe:quantity: quantity:BACK_CTS_RAW
  sml:output:
  - sml:name: cts_mod_corr
    swe:quantity: quantity:CTS_MOD_CORR_LEVEL2
  sml:components:
  - "@type": sml:AggregateProcess
    sml:name: cts_moderation
    sml:input:
    - sml:name: cts
      swe:quantity: quantity:BACK_CTS_RAW
    sml:output:
    - sml:name: cts_mod
      swe:quantity: quantity:CTS_MOD_RAW
    sml:components:
    - "@type": sml:SimpleProcess
      sml:name: bcg_var_removal
      rdfs:label: Removal of background variability
      sml:input:
      - sml:name: cts
        swe:quantity: quantity:BACK_CTS_RAW
      sml:output:
      - sml:name: cts
        swe:quantity: quantity:BACK_CTS_RAW
    - "@type": sml:SimpleProcess
      sml:name: atm_eff_removal
      rdfs:label: Removal of atmospheric effects
      sml:input:
      - sml:name: cts
        swe:quantity: quantity:BACK_CTS_RAW
      - sml:name: pa
        swe:quantity: quantity:PA_LEVEL2
      - sml:name: rh
        swe:quantity: quantity:RH_LEVEL2
      sml:output:
      - sml:name: cts_mod
        swe:quantity: quantity:CTS_MOD_RAW
    sml:connections:
    - "@type": sml:Link
      sml:source: inputs/cts
      sml:destination: components/bcg_var_removal/inputs/cts
    - "@type": sml:Link
      sml:source: components/bcg_var_removal/outputs/cts
      sml:destination: components/atm_eff_removal/inputs/cts
    - "@type": sml:Link
      sml:source: components/atm_eff_removal/outputs/cts_mod
      sml:destination: outputs/cts_mod
  - "@type": sml:SimpleProcess
    sml:name: snow_days_removal
    rdfs:label: Removal of snow days identified from radiation albedo
    sml:input:
    - sml:name: cts_mod
      swe:quantity: quantity:CTS_MOD_RAW
    sml:output:
    - sml:name: cts_mod
      swe:quantity: quantity:CTS_MOD_RAW
  - "@type": sml:SimpleProcess
    sml:name: gamma_corr
    rdfs:label: Gamma correction using site-specific factor
```

```yaml
    sml:input:
    - sml:name: cts_mod
      swe:quantity: quantity:CTS_MOD_RAW
    sml:output:
    - sml:name: cts_mod_corr
      swe:quantity: quantity:CTS_MOD_CORR_LEVEL2
  sml:connections:
  - "@type": sml:Link
    sml:source: inputs/cts
    sml:destination: components/cts_moderation/inputs/cts
  - "@type": sml:Link
    sml:source: components/cts_moderation/outputs/cts_mod
    sml:destination: components/snow_days_removal/inputs/cts_mod
  - "@type": sml:Link
    sml:source: components/snow_days_removal/outputs/cts_mod
    sml:destination: components/gamma_corr/inputs/cts_mod
  - "@type": sml:Link
    sml:source: components/gamma_corr/outputs/cts_mod_corr
    sml:destination: outputs/cts_mod_corr
- "@type": sml:SimpleProcess
  sml:name: cts_to_vwc
  rdfs:label: Neutron count to VWC conversion using site-specific calibration data
  sml:input:
  - sml:name: cts_mod_corr
    swe:quantity: quantity:CTS_MOD_CORR_LEVEL2
  sml:output:
  - sml:name: vwc
    swe:quantity: quantity:LEVEL2_VWC_GAMMA_1DAY
sml:connections:
- "@type": sml:Link
  sml:source: inputs/cts
  sml:destination: components/cts_correction/inputs/cts
- "@type": sml:Link
  sml:source: components/cts_correction/outputs/cts_mod_corr
  sml:destination: components/cts_to_vwc/inputs/cts_mod_corr
- "@type": sml:Link
  sml:source: components/cts_to_vwc/outputs/vwc
  sml:destination: outputs/vwc
```

11.7 Water quality data

```yaml
# This example shows the potential encoding of the lab analysis of the concentration of soluble reactive phosphorus
# in a sample taken, prepared and stored using specific procedures.
# In the example, we use complex property model to encode the nature of the determinand; SensorML to encode
# sampling, preparation and analysis procedures; and SOSA/SSN to encode the act of sampling and laboratory
# observation.

# Complex property. Here, it doesn't carry any details of the process. Practices differ here: for instance,
# BODC P01 Chemical Entity Parameter Code Builder supports (and encourages) the inclusion of the process details.
- "@id": property:SRP
  "@type": cpm:ObservableProperty
  rdfs:label: Soluble reactive phosphorus (µg l-1-P)
  cpm:objectOfInterest: substance:phosphorus
  cpm:constraint: qualifier:soluble_reactive
  cpm:property: property:concentration
  cpm:uom: unit:µg_per_l

# Description of the sampling, sample preparation and handling process. Here, modelled as SensorML processes.
- "@id": procedure:sample_prep_for_SRP
  "@type": sml:AggregateProcess
  sml:components:
    - "@type": sml:SimpleProcess
      sml:typeOf: procedure:sampling
      dct:description: Bulk samples are taken from the main flow of each river on Monday or Tuesday of each week
    - "@type": sml:SimpleProcess
      sml:typeOf: procedure:filtration
```

```
      dct:description: Subsamples were filtered immediately in the field through a 0.45 µm Whatman WCN membrane
    - "@type": sml:SimpleProcess
      sml:typeOf: procedure:storage
      dct:description: On return to the laboratory, all samples were stored in the dark at 4C, prior to analysis

# The act of sampling, its result is a sample. Here, it's also used to model consequent sample handling.
- "@type": sosa:Sampling
  sosa:hasResult: sample:643206886
  sosa:resultTime: 2019-09-01T10:46:00Z
  sosa:hasFeatureOfInterest: site:THE_CUT
  sosa:hasUltimateFeatureOfInterest: river:THAMES
  sosa:usedProcedure: procedure:sample_prep_for_SRP

# Observation performed on the sample in a lab. Note the advantage of SOSA/SSN: procedure is separated from sensor
# and so it can be a generic method. The sensor capabilities carry the information on the calibration, limit of
# detection etc.
- "@type": sosa:Observation
  sosa:hasFeatureOfInterest: sample:643206886
  sosa:observedProperty: property:SRP
  sosa:sensor: sensor:Seal_Auto_Analyser_3_6512
  sosa:usedProcedure: procedure:phosphomolybdenum_blue_colorimetry_Murphy_Riley
  sosa:resultTime: 2019-09-02T09:15:00Z
  sosa:hasSimpleResult: 351.0
```

11.8 Complex results: Temperature profile

```
# Data measured by profile sensors could be expressed as a complex -
# coverage result of a single observation, which aims to describe
# values of a property (e.g. temperature) in different places (and
# times - see time series and spatiotemporal) of a single feature-of-interest.

# This keeps things more compact and does not require definition of
# specific properties or FoIs for different depths.

---
- "@type": inspireOm:ProfileObservation
  om:procedure: sensor:Hukseflux_STP01_6872
  om:observedProperty: property:STP_TSOIL_RAW  # Soil temperature profile (raw)
  om:featureOfInterest: # Spatial sampling feature of soil at the location of the profile sensor
    "@type": sf:SamplingCurve
    rdfs:label: soil at location of STP on CHIMN
    sf:sampledFeature: sweet:realmSoil
    sf:relatedSamplingFeature: site:CHIMN # this would also define coordinate reference system
    sf:shape: location:STP
  om:phenomenonTime : 2019-09-06T13:30:00.000Z
  om:result:
    "@type": cv:ReferenceableGridCoverage

    # Here, we'd specify how the grid is laid on the Feature of Interest geometry
    gml:domainSet:
      "@type": csml:ReferenceableGridByVectors
      # gml:gridEnvelope: ...
      # dimension: 1
      # origin: ...
      # axisLabels: x y z
      # sequenceRule: ... to state how the sequence encodes the data
      csml:generalGridAxis:
        csml:offsetVector: 0 -1 0 # this is y=depth
        qudt:unit: unit:CM
        csml:coefficients: # with which the offset vector is multiplied
          - 2.0
          - 5.0
          - 10.0
          - 20.0
          - 50.0
```

```yaml
    # Here, we specify the sequence of values. Their mapping to the domainSet is defined within the domainSet
    gml:rangeSet:
      qudt:unit: unit:DEG_C
      gml:dataBlock:
        - 21.2
        - 21.0
        - 20.9
        - 20.7
        - 20.3
      # ...

# The result could also be represented using SWE Common data model Arrays (see section B.2 of SWE Common)
```

11.9 Complex results: Wind

```yaml
# Wind direction as a single observation with complex result.
# Here, we use SWE Common data structures but O&M also suggests the
# use of gm:Object (from ISO/TS 19107:2003) or Record (from ISO/TS 19103:2005).
# Declaring the vector shape and unit of measure could just be done
# once and result quantities could just reference that definition.
---
- "@type": om:Measurement
  om:procedure: sensor:Gill_WindMaster_3D_Sonic_Anemometer_1568
  om:observedProperty: property:WIND_DIRECTION_RAW
  om:featureOfInterest: site:CHIMN
  om:phenomenonTime: 2019-09-06 13:30:00 UTC
  om:resultTime: 2019-09-06 13:50:00 UTC
  om:result:
    "@type": swe:Vector
    swe:coordinates:
    - swe:axisID: x
      swe:value: 7.4
      swe:uom: unit:mps
    - swe:axisID: y
      swe:value: 10.1
      swe:uom: unit:mps
    - swe:axisID: z
      swe:value: 5.3
      swe:uom: unit:mps


# More complex case with mean and stdev for each value.
---
- "@type": om:Measurement
  om:procedure: sensor:Gill_WindMaster_3D_Sonic_Anemometer_1568
  om:observedProperty: property:WIND_DIRECTION_RAW
  om:featureOfInterest: site:CHIMN
  om:phenomenonTime: 2019-09-06 13:30:00 UTC
  om:result:
    "@type": swe:Vector
    swe:coordinates:
    - swe:axisID: x
      swe:value:
        "@type": swe:DataRecord
        mean: 7.4
        stdev: 1.3
      swe:uom: unit:mps
    - swe:axisID: y
      swe:value:
        "@type": swe:DataRecord
        mean: 10.1
        stdev: 3.2
      swe:uom: unit:mps
    - swe:axisID: z
      swe:value:
        "@type": swe:DataRecord
```

```
        mean: 5.3
        stdev: 0.9
      swe:uom: unit:mps
```

11.10 Complex results: TDT results coverage

Data measured by TDT sensors could be expressed as a complex - coverage result of a single observation made by a system of sensors, which aims to describe values of a property (e.g. temperature) in different places (and times - see time series and spatiotemporal) of a single feature-of-interest - site in this case. This keeps things more compact and does not require definition of specific properties or FoIs for TDT1-10.

A disadvantage here is that values are not linked to the specific TDT they were obtained with. So if one of the TDTs gets replaced, the information is not included in the observations - further provenance information will need to be supplied. Of course, the system of TDT sensors would normally be declared elsewhere.

```
---
- "@type": om:DiscretePointCoverageObservation
  om:procedure:
    "@type": sml:PhysicalSystem
    rdfs:label: TDTs at CHIMN
    sml:featuresOfInterest: site:CHIMN
    sml:components:
    - sensor:Acclima_Digital_TDT_6476
    - sensor:Acclima_Digital_TDT_9941
  om:observedProperty: property:TDT_SOILEC_RAW
  om:featureOfInterest: site:CHIMN
  om:phenomenonTime: 2019-09-06 13:30:00 UTC
  om:result:
    "@type": cv:DiscretePointCoverage
    cv:elements:
    - "@type": cv:GeometryValuePair
      cv:geometry: location:TDT1  # "Reusable" position relative to a coordinate reference point for the site
      cv:value: 9.3
    - "@type": cv:GeometryValuePair
      cv:geometry: location:TDT2
      cv:value: 9.7
```

The same example, now extended so that the coverage is now spatio-temporal - different values for different times. Temporal elements could also be time intervals, which will make it easier to understand which interval temporal aggregations are performed on (preceding / succeeding).

```
- "@type": om:Measurement
  om:procedure:
    "@type": sml:PhysicalSystem
    rdfs:label: TDTs at CHIMN
    sml:featuresOfInterest: site:CHIMN
    sml:components:
    - sensor:Acclima_Digital_TDT_6476
    - sensor:Acclima_Digital_TDT_9941
  om:observedProperty: property:TDT_SOILEC_RAW
  om:featureOfInterest: site:CHIMN

  # This will probably be defined more structurally
  om:phenomenonTime: 2019-09-06T00:00:00.000Z - 2019-09-07T00:00:00.000Z

  om:result:
    "@type": cv:DiscretePointCoverage
```

```yaml
    cv:elements:
    - "@type": cv:GeometryValuePair
      cv:geometry:
        cv:spatialElement: location:TDT1
        cv:temporalElement: 2019-07-06 00:30:00 UTC
      cv:value: 9.3
    - "@type": cv:GeometryValuePair
      cv:geometry:
        cv:spatialElement: location:TDT2
        cv:temporalElement: 2019-07-06 00:30:00 UTC
      cv:value: 9.7
    - "@type": cv:GeometryValuePair
      cv:geometry:
        cv:spatialElement: location:TDT1
        cv:temporalElement: 2019-07-06 01:00:00 UTC
      cv:value: 9.4
    - "@type": cv:GeometryValuePair
      cv:geometry:
        cv:spatialElement: location:TDT2
        cv:temporalElement: 2019-07-06 01:00:00 UTC
      cv:value: 9.7
    # ...
```

11.11 Complex results: Timeseries

```yaml
# Here, we demonstrate the use of TimeseriesML to represent a series of
# rain gauge measurements. The spec allows a much wider range of metadata than
# what we demonstrate here, particularly useful in case of cumulative time series
# or when combining data from existing series e.g. to interpolate.

# Measurement (subclass of om:Measurement)
- "@type": tsml:MeasurementTimeseriesTvpObservation
  om:procedure: sensor:OTT_Pluvio_6182
  om:observedProperty : property:PRECIPITATION_RAW
  om:featureOfInterest : site:CHIMN
  om:phenomenonTime :
    tm:begin: 2019-09-06T00:00:00.000Z
    tm:end: 2019-09-07T00:00:00.000Z
  om:resultTime : 2019-09-06T13:50:00.000Z
  om:result :

    # Timeseries
    "@type": tsml:MeasurementTimeseriesTvp
    tsml:metadata:
      "@type": tsml:MeasurementTimeseriesMetadata
      tsml:temporalExtent:
      tsml:baseTime: 2019-09-06T00:00:00.000Z
      tsml:sampledMedium: sweet:atmoWeather
      tsml:intendedObservationSpacing: interval:30min
      tsml:cummulative: false

    tsml:elements:
      # Elements encoded as time-value pairs
      - "@type": tsml:MeasureTimeValuePair
        tsml:geometry:
          tsml:temporalElement: 2019-07-06T00:30:00.000Z
        tsml:value: 0.21
        tsml:metadata:
          "@type": tsml:MeasurementPointMetadata
          tsml:interpolationType: tsmlIntCode:PrecTotal
          tsml:qualifier:
            - err:SENSOR_FAULT
    # - ...
```

11.12 Dataset annotation: Monitored properties

```
# We're presenting a dataset containing observations for a couple of monitored properties
# that describe its content: what was measured, where and using what process.

---
- "@id": dataset:814b2005-cd82-49bf-b2f7-5c82f61d08f4
  "@type": mp:MonitoringDataset
  mp:storesValuesFor:
    - "@type": mp:MonitoredProperty
      rdfs:label: Soil temperature profile (10 cm depth) (raw) at CHIMN
      mp:monitoredFeature: site:CHIMN
      mp:monitoredObservableProperty: property:STP_TSOIL10_RAW
      mp:monitoringProcess: sensor:Hukseflux_STP01_6872

    - "@type": mp:MonitoredProperty
      rdfs:label: Soil temperature profile (20 cm depth) (raw) at CHIMN
      mp:monitoredFeature: site:CHIMN
      mp:monitoredObservableProperty: property:STP_TSOIL20_RAW
      mp:monitoringProcess: sensor:Hukseflux_STP01_6936

    # ...

  member:
    - "@type": om:Observation
    # ...
```

12 Appendix B Diagrams

12.1 COSMOS Data Model

Complete existing metadata model is depicted in figure 27 below.

- Thick lines denote main classes and relationships between them.
- green classes = meta: database tables and columns
- yellow classes = existing tables with data
- white classes = empty existing tables (no data available)
- gray classes = looks like enumerations / code lists but without relevant tables
- black solid lines = identified references to other classes of any kind
- gray dashed lines = potential (not currently managed) relationships
- Site time series class is generated - solid lines denote its members links (just like solid black lines) and dashed lines what tables it is derived from.

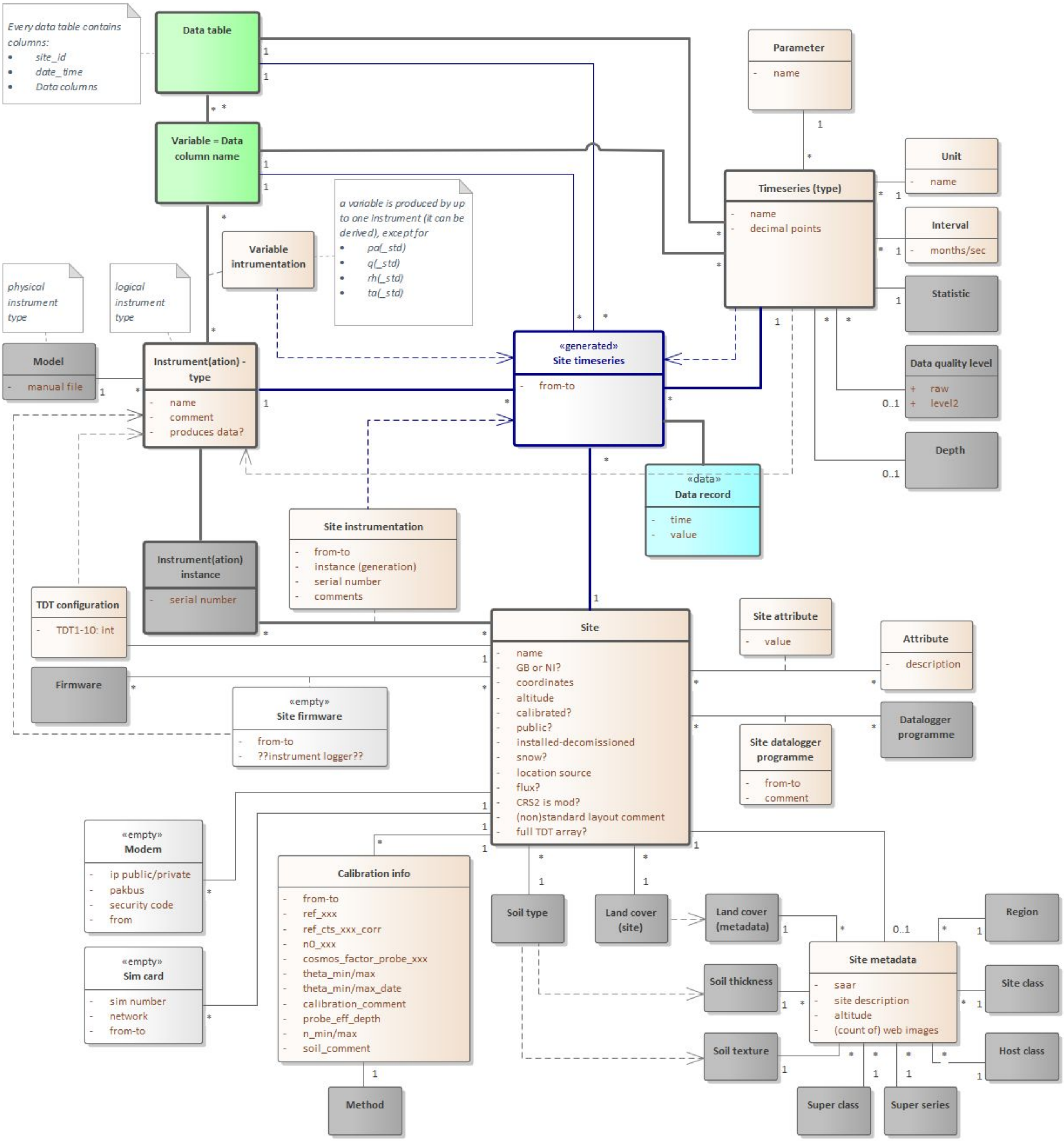


Figure 27 COSMOS-UK high-level data model

12.2 NRFA Data model

Detailed conceptual model of the National River Flow Archive Colours is shown below. Colors determine groups of related information:

- Green, blue = Station and catchment information, respectively
- White = Gauged daily flow data - uniform structures for archived and live data
- Turquoise = Archived daily flow data
- Yellow = Live (EA) daily flow data
- Red = Peak flow data
- Gray = Photo gallery
- Thick black lines indicate main entities and relationships between them
- Other black lines indicate other discovered links
- Gray lines indicate assumed links

Majority of the reference data is already described on https://nrfa.ceh.ac.uk/, e.g.:

- Measuring authority: https://nrfa.ceh.ac.uk/measuring-authorities
- Hydrometric area: https://nrfa.ceh.ac.uk/hydrometric-areas
- Category / Network: https://nrfa.ceh.ac.uk/nrfa-categories-networks

Complete list is included in more detailed model in the attached Enterprise Architect model file as descriptions of individual classes.

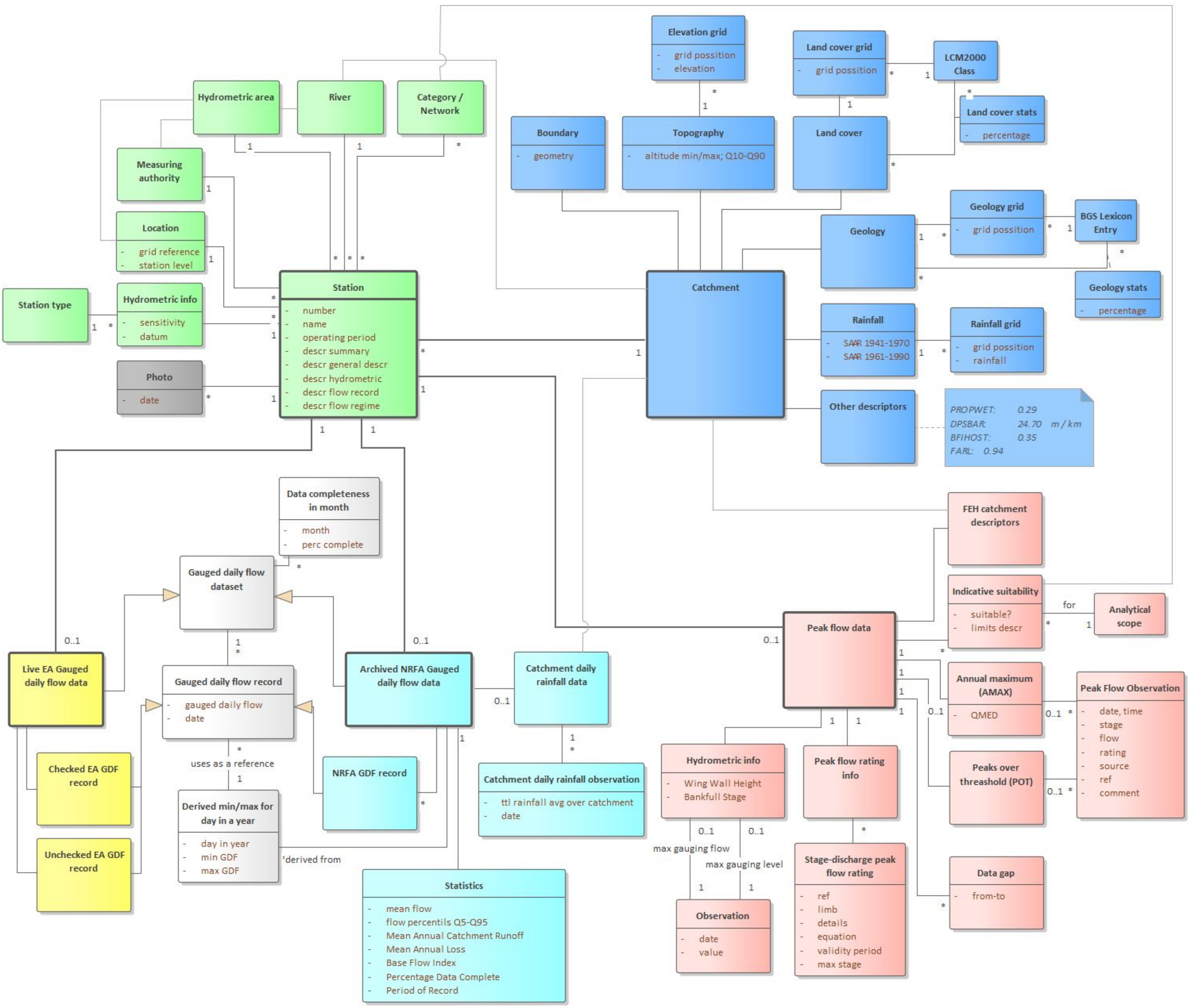


Figure 28 NRFA high-level data model