TT-AvXML: UML derivation from TDCF

WMO TT-AvXML [M1]

Outline of discussions and agreements; notes compiled by Jeremy Tandy (Met Office) and are supplementary to the official meeting minutes.

ICAO requirements

ICAO MET information Logical Data Model shall form part of ICAO AIRM (Aeronautical Information Reference Model) – availability of AIRM expected 2012Q3.

ATM requirements for MET information are 'still evolving' – irrespective of progress from NextGEN and SESAR there is no global consensus for operations. ICAO MARIE-PT is expected to develop these for publication at ICAO Divisional Meeting mid-2014.

In lieu of future MET information requirements, urgency is required on the specification of XML/GML Schema for TAF, METAR/SPECI and SIGMET (VA SIGMET, TC SIGMET & WS SIGMET) to support Amendment 76 to ICAO Annex 3 (WMO No. 49): (*for States in a position to do so*) permitting bilateral exchange of OPMET data via XML.

There is no commitment from ICAO to requirements beyond TAF, METAR/SPECI and SIGMET.

Aviation community seek to harmonise data exchange technology around XML to reduce overall cost-base of ATM / SWIM system; BUFR seen as 'odd-ball' that requires specific technology stacks to implement.

Target delivery date for XML schema to support Amendment 76: July 2012.

ICAO shall own the MET information Logical Data Model from which the XML schema is derived.

Metadata shall be expressed using ISO 19115 profile; work-in-progress within OGC Aviation DWG, custodianship under review by ICAO. Implications for TT-AvXML are not yet known.

Meteorological information Logical Data Model: derived from TDCF

Logical Data Model: ISO/TC211 parlance = 'Application Schema'. ISO 19109 provides rules for Application Schema.

WXXM2.0 shall be baseline input to ICAO MET information Logical Data Model. WXXM2.0 shall be based on ISO 19156 Observations and Measurements. Concerns were raised concerning the inclusion of US-regional variations within WXXM.

WMO Logical Data Model seeks to build upon ISO 19156 Observations and Measurements and CSML 3 to achieve compatibility with INSPIRE Annex 3 Data Specifications, Unidata CDM (data model for netCDF) and CF-point conventions. WXXM2.0 is unlikely to have adopted CSML 3 due to inadequacies in documentation. The adoption of CSML 3 'patterns' within INSPIRE Data Specification AC/MF was noted.

Following proposal of Document 3.2.3 [BUFR-lite (Jeremy Tandy, Met Office)[1]] WMO TT-AvXML agreed the following:

- Sequences expressed in BUFR Table-D play a similar role to Classes within a Logical Data Model; collecting groups of properties together.
- It is conceivable to automatically generate Classes from BUFR Table-D and associated entities from BUFR Table-B. Care should be taken when converting BUFR Table-D Sequences to Classes – especially where 'replication sequences' are used. The use of other BUFR Table-C modifiers is also likely to be problematic.

[1] The BUFR-lite paper is based on discussions with the Met Office team; special thanks to Gil Ross

- Treating BUFR Table-D Sequences as Classes is inconsistent with the current 'serial' processing model employed within BUFR (modal behaviour). An approach where BUFR Table-D Sequences are treated as Classes requires BUFR to adopt a 'lexical' processing model. Enrico Fucile (ECMWF) was keen to adopt this change within the forthcoming BUFR Edition (BUFR 5).
- 'Proto-Classes' derived from the Code-Tables must be compared against existing definitions from ISO/TC211 Logical Data Models (e.g. ISO 19103 Conceptual Schema Language, ISO 19107 Spatial Schema, ISO 19108 Temporal Schema, ISO 19111 Spatial Referencing by Coordinates , ISO 19115 Metadata, ISO 19123 Coverages and ISO 19156 Observations and Measurements).
- Human-readable 'short-names' will be required for each 'proto-class'.
- Adoption of base Classes from ISO/TC211 and 're-factoring' of replication sequences from Table-D may require amendment to existing Table-D Sequences to ensure alignment between Code-Tables and Logical Data Model. Furthermore, the human readable 'short-names' may be proposed for inclusion within the Code-Tables themselves. Such changes can be proposed to IPET-DRC at the next meeting (see below).

Initially, constraints expressed in BUFR documentation shall be expressed as annotations in the Logical Data Model. For future iterations, it is anticipated that such free-text annotations will be converted to formal OCL (Object Constraint Language) statements that can then be automatically encoded as Schematron rules.

TDCF does not support concept of inheritance (Generalisation / Specialisation of Classes) – such relationships may be incorporated within WMO Logical Data Model.

It is essential that the WMO Logical Data Model be maintained. A clearly defined process must be agreed to keep the Logical Data Model synchronised with the Code-Tables.

Special care must be taken if WMO decides to maintain BUFR Table-D Sequences for entities managed within the ICAO MET information Logical Data Model (TAF, METAR/SPECI and SIGMET); such entities must not be republished within the WMO Logical Data Model – their presence in the Code-Tables merely enables transliteration of Aviation XML to BUFR-encoding. Candidate proposal: create a new Table-D 'Class' for these objects & ensure that this Class is not converted.

Distributed governance: importing WMO Logical Data Model 'generic weather' packages

ICAO MET information Logical Data Model will build on many generic meteorological concepts. Such generic 'weather' Classes and Concepts will be re-used across other domains. These entities shall be managed by WMO. WMO shall publish a 'Generic MET information' Logical Data Model (WMO Logical Data Model) that contains generic meteorological definitions for use in multiple domains / industries.

ICAO MET information Logical Data Model ***should*** import Packages as necessary from WMO Logical Data Model.

WMO data specification standards shall be treated identically by ICAO to those published by ISO/TC211. The implication is that WMO will manage its standards (including the Logical Data Model) with equivalent rigour to ISO/TC211. *WMO data specifications may be published with ISO document numbers.*

Jeremy Tandy (Met Office, representing WMO CBS) has been invited to participate in ICAO MARIE-PT to ensure that the relationship between ICAO MET information Logical Data Model and WMO Logical Data Model is mutually supportive.

Definitions shall be specific to a specific namespace; note that 'freezing' and 'visibility' definitions differ between ICAO and WMO … *debate required to determine which definition is used in ICAO MET information Logical Data Model.*

For interim release (July 2012) ICAO Met information Logical Data Model shall be self-contained:

- Over-arching AIRM will not be ready until 2012Q3; local definitions for Aerodrome, Runway, Runway Segment, Flight Information Region (FIR) and Airspace Segment Feature Types will be provided within the ICAO MET information Logical Data Model.
- ICAO MET information model shall include *copy* of WMO Logical Data Model packages

Conversion to XML/GML Schema

WXXS: Weather eXchange Schema; an XML/GML Schema.

Considerations regarding Efficient XML Interchange (EXI) are out of scope for WMO TT-AvXML.

ISO19136 (GML) provides rules for conversion of Application Schema to XML/GML Schema.

WMO shall own WXXS[2]; the *Physical Data Model*.

WXXS shall be automatically derived from ICAO MET information Logical Data Model.

Concerns were noted regarding verbosity of GML3.2.1 (ISO 19136); SESAR is investigating whether GML2.4 provides adequate complexity.

Expert representation from WMO CBS within ICAO MARIE-PT should ensure that the ICAO MET information Logical Data Model does not include approaches that adversely affect the implementation of the Physical Data Model.

A clear benefit of the Model-Driven approach is that data-exchange standards become 'robust' to changes in underlying technology. The Logical Data Model provides the authoritative content standard (semantic definitions). Different physical encodings can be automatically derived from the Logical Data Model. Thus as XML is replaced by a newer technology (JSON?) direct transliteration between encodings is possible. Furthermore, if BUFR Table-D Sequences are reverse-engineered from the ICAO MET information Logical Data Model, it will be possible to achieve a lossless (bi-directional) transliteration between XML/GML and BUFR-encoding. Whilst there is no requirement from ICAO to maintain BUFR-encodings of TAF, METAR/SPECI and SIGMET for international data exchange, some centres may choose BUFR as a compact encoding for archival.

Automated tooling for converting (annotated) UML to XML/GML Schema: Fullmoon (CSIRO developed, released as open source).

NCAR (FAA NextGen project team) are (currently) overhauling Fullmoon so that the XML/GML schema produced is as expected & does not require manual intervention.

B L Choy (Hong Kong Observatory) offered to lead on the Application Schema – XML/GML conversion & associated Fullmoon deployment & configuration.

Fullmoon produces XML Schema packages that reference existing authoritative XML Schema definitions; one only needs to publish XML Schema for the parts of the model you manage. Implication: WMO must pre-publish XML Schema derived from WMO Logical Data Model for inclusion in WXXS.

Schematron rules will be established to express the constraints from the BUFR documentation; publication anticipated _AFTER_ the interim July 2012 release.

Outline Process:

1. WMO _LOGICAL DATA MODEL_ shall be inferred from the TDCF and maintained in synchronisation
2. WMO shall derive XSD from WMO _LOGICAL DATA MODEL_ using an automated process such as Fullmoon
3. ICAO MET information _LOGICAL DATA MODEL_ should import packages from WMO _LOGICAL DATA MODEL_ as appropriate
4. ICAO MET information _LOGICAL DATA MODEL_ shall compose Classes and Datatypes imported from WMO _LOGICAL DATA MODEL_ into (new) locally defined Classes[3]

[2] Or rather – the ICAO MET information XML schema (e.g. Physical Data Model)

[3] These locally defined Classes (e.g. for TAF, METAR/SPECI & SIGMET) shall be under ICAO governance - and shall not be included in the WMO _LOGICAL DATA MODEL_. The derivation of the WMO _LOGICAL DATA MODEL_ from TDCF must be mindful of this requirement

5. On behalf of ICAO, WMO shall derive XSD from ICAO MET information _LOGICAL DATA MODEL_ - importing WMO XSD as necessary (WMO owns WXXS)

Deriving a UML model from TDCF

Class-names:

Assumption: BUFR/CREX Codes from Table B and Table D shall be converted into UML Classes

Each UML Class will need to be given an appropriate name.

Inclusion of the BUFR Table designator and Class designator will add unique scope. Furthermore, the Code number from the BUFR Table shall be included in the Class-name to further disambiguate; e.g. B12-003_DewpointTemperature.

For example: there are several 'dew-point temperature' measures in Table B Class 12 - each of which have different encoding details and/or units for BUFR. For example (from Table B Class 12):

F X Y	Element name	Unit	Scale	Reference value	Data width (bits)
0 12 003	Dew-point temperature	K	1	0	12
0 12 024	Dew-point temperature	C	0	-99	8
0 12 103	Dew-point temperature	K	2	0	16

Effectively, [0 12 103] can encode Dew-point temperature to 1/100th degree (expressed in Kelvin), [0 12 003] with precision 1/10th degree (expressed in Kelvin), whilst [0 12 024] is limited to precision of 1 degree (expressed in Celsius). This will result in the creation of 3 different Classes:

- B12-003_DewpointTemperature
- B12-024_DewpointTemperature
- B12-103_DewpointTemperature

Modelling BUFR Table B entities

The Class-name shall serve as a primary label.

If the UML model is ever to replace the WMO TDCF Code Tables, it is essential that the UML model can hold the additional information expressed in the Code Tables (BUFR / CREX Table B):

- Code number
- Element name
- BUFR:
    - Scale
    - Reference value
    - Data width (bits)
- CREX:
    - Scale
    - Data width (characters)
- Status

Following the practices adopted within ISO/TC 211 for capturing additional information for deriving GML/XML encodings from the UML model using the FullMoon processing software, we propose to develop an Extended UML Profile to support WMO TDCF encoding. This profile will provide a set of tagged values that can be used to capture this additional information. It is also imperative to ensure the correct ordering of elements within the BUFR/CREX encoding. A tagged value 'sequenceNumber' shall be used for this purpose.

Thus an extended UML metamodel for WMO TDCF might include the following tagged value options:

- sequenceNumber *(the ordering may differ from XML encoding - but it is unlikely)*
- Table reference (FXY)[4]
- ElementName
- BUFR_Unit
- BUFR_Scale
- BUFR_ReferenceValue
- BUFR_DataWidth
- CREX_Unit
- CREX_Scale
- CREX_DataWidth
- Status

ASIDE: expressing precision information

The combination of Scale, Reference value and Data width imposes a limit on the precision of information that can be exchanged in BUFR and CREX. Other encodings, such as XML, are unlikely to share such limitations imposed by the physical model. However, it is worth noting that the WMO TDCF values have been chosen such that the precision of the data is sensible. Thus it may make sense to use this information to infer a statement about the precision of data that can be captured in each element. Such information could be expressed in:

- tagged values (as indicated above)

- UML note[5] (these may be harvested during the conversion to GML/XML schema to provided additional contextual information in the XML Schema document)

- contraint (a constraint may be converted into schematron rules; this might be useful in processing XML encoded data)

For the initial phase, none of these options have been taken due to the need for expediency.

Measures

Take the following example from Table B Class 12 'Temperature':

Code	Element name	BUFR				CREX			Status
		Unit	Scale	Reference value	Data width (bits)	Unit	Scale	Data Width (characters)	
0 12 024	Dew-point	C	0	-99	8	C	0	2	Operational

[4] Including the BUFR/CREX reference is is important because this should provide a mechanism that enables traceability from the UML model to the BUFR codes. For example, Class ISO19108:TM_Instant is used to express a time instant, we can include a tagged value to relate this Class to the BUFR Table D Sequence describing date-time instant.

[5] e.g. supplemental information capture about the Class - rather than a Comment that appears within a UML diagram

METAR Sequence contains Code [0 12 024] 'Dew-point temperature'; this indicates that the **value** decoded from a particular 8-bit width fragment of the binary file shall be 'dew-point temperature' defined in unit of measure: Celsius. For info, the decode process is:

bin[01101110] = dec[110]

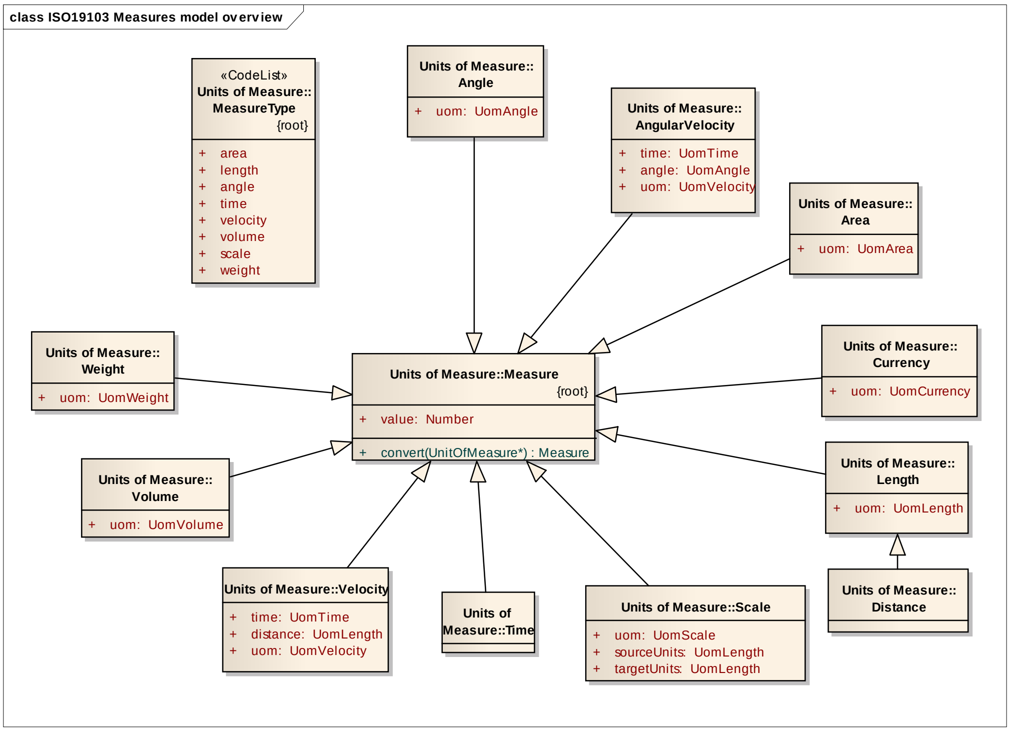
(in BUFR, from left-to-right, bit 1 is most significant)

{original value} = ({encoded value} + {reference}) / 10^{scale}

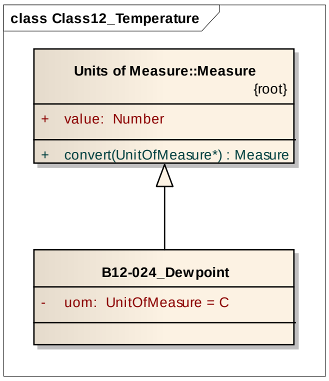
Taking the appropriate values from the BUFR Code-table we can deduce that the dew-point temperature recorded is 11 ^{o}C:

(110 + (-99)) / 10^0 = 11

ISO 19103 'Conceptual schema language' provides a series of Measure Classes already - along with a UnitOfMeasure Class and further sub-Classes thereof:



BUFR Table B Class 12 code [0 12 024] 'Dew-point temperature' is another type of Measure. The methodology inferred from ISO 19103 is that each specific type of Measure shall specialise Class Measure. Clearly the unit of measure is essential to ensure that the value can be interpreted. Thus the Class representing this code should (at a minimum) include the following attributes:



Note: BUFR and CREX encodings specify the unit of measure used for a given Code. This unit of measure regularly varies between the BUFR and CREX encoding of the same Code entity. Within the Logical Data Model, it is clear that this Class _MUST_ provide information about the unit of measure in order to interpret the value - without a unit of measure, the value is meaningless. The Logical Data Model _MAY_ provide a default unit of measure; this can be overridden in BUFR or CREX encoding as necessary.

All Codes in the following BUFR Table B Classes will have similar requirements:

- Class 10 - Non-coordinate location (vertical)
- Class 11 - Wind and turbulence
- Class 12 - Temperature
- Class 13 - Hydrographic and hydrological elements
- Class 14 - Radiation and radiance
- Class 15 - Physical/chemical constituents
- Class 19 - Synoptic features
- Class 20 - Observed phenomena
- Class 21 - Radar data
- Class 22 - Oceanographic elements
- Class 23 - Dispersal and transport
- Class 24 - Radiological elements
- Class 25 - Processing information
- Class 26 - Non-coordinate location (time)
- Class 27 - Non-coordinate location (horizontal - 1)
- Class 28 - Non-coordinate location (horizontal - 2)
- Class 30 - Image
- Class 33 - Quality Information
- Class 35 - Data monitoring information
- Class 40 - Satellite data

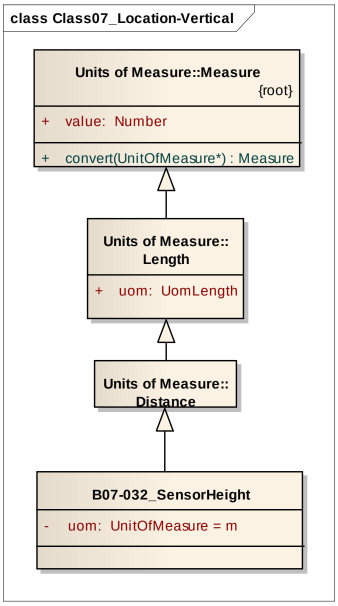
Class 31 - Descriptor operator qualifiers] is excluded from this list as these entities appear to be concerned with encoding issues within the physical BUFR format.

Generalised Coordinates

Generalised Coordinate Classes (Class 01 'Identification', Class 02 'Instrumentation', Class 04 'Location (time)', Class 05 'Location (horizontal - 1)', Class 06 'Location (horizontal - 2)' and Class 07 'Location (vertical)') should not routinely derive from Measure. However, exceptions do exist, for example:

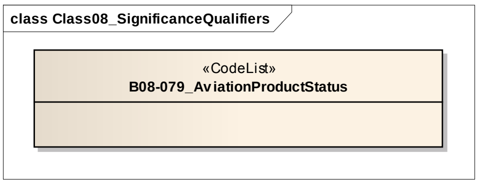
- Class 05 'Location (horizontal -1)' [0 05 021] 'Bearing or azimuth'
- Class 07 'Location (vertical)' [0 07 030] 'Height of station ground above mean sea level'
- Etc.

These elements shall specialise Measure (or sub-Classes thereof) as appropriate.



Significance Qualifiers

Class 08 'Significance qualifiers' are also treated differently. These are _NOT_ measures; they are routinely used to Qualify a measure or some other value. Early-stage analysis suggests that each Significance Qualifier entity references a CodeList with a managed set of Terms. As such, Significance Qualifiers are stereotyped «CodeList» (see below for more details).



Code-tables

Many of the Codes defined within these BUFR/CREX Classes specify 'unit' as "Code table".

Each of these Code-tables shall be managed independently of the Logical Data Model as CodeLists, enabling terms within each CodeList to be amended without impacting the Logical Domain Model.

Following current best practice on representing CodeLists / ConceptSchemes in UML Application Schema (Cox[6], Lutz[7] et al), each CodeTable shall be modelled with a sterotyped «CodeList» Class. The following rules are taken from INSPIRE best-practice documentation:

---

[6] Simon Cox, CSIRO

[7] Michael Lutz, JRC

- All values in a CodeList shall be durable (long-term persistent) – even when deprecated, retired or superseded
- «CodeList» Classes shall be shown empty on UML diagrams; the values of CodeLists shall not be included within the UML Model. Some example terms may be included in a note on the Class diagram; this is purely informative.
- «CodeList» Classes shall carry some kind of indication of where the list is maintained and how it should be access - possibly using tagged values[8]

The «CodeList» stereotype should be taken to imply a UML Realization of an «Interface» Class carrying operations corresponding to CodeList behaviour; e.g.

+validateCode(CharacterString):boolean

+getList():Set

Each BUFR Table B Code that references a CodeList (e.g. unit = 'Code table') shall be stereotyped «CodeList». The CodeList and the values therein will be published as web-accessible Resources - each with a specific URI[9].

For example: BUFR Code [0 13 040] 'Surface flag'

«CodeList» B13-040_SurfaceFlag

Code figure	Description
0	Land
1	Reserved
2	Near coast
3	Ice
4	Possible ice
5	Ocean
6	Coast
7-14	Reserved
15	Missing value

XML example of using a CodeList (this is correct for GML Dictionaries - not sure how it works if the target is a SKOS Concept Scheme / SKOS Collection):

```
<B13-040_SurfaceFlag codeList="http://{hostname}/{path}/Code+Flag/B_13_040"
codeListValue="http://{hostname}/{path}/Code+Flag/B_13_040/0">Land</B13-040_SurfaceFlag>
```

Flag-tables

Where the BUFR/CREX Classes specify 'unit' as "Flag table" a similar approach to Code-tables can be used - albeit allowing multiple 'flags' to specified instead of a single code-values.

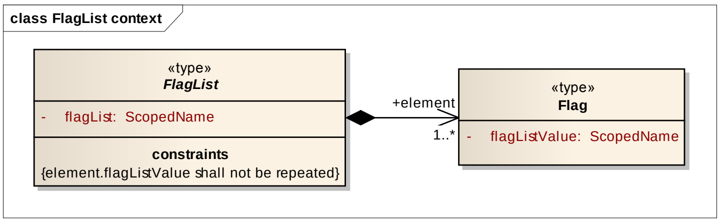
Unfortunately, this will not prevent inappropriate combinations of flags being set. However, the existing BUFR/CREX encodings do not prevent this situation either.

This behaviour is modelled following a similar approach to CodeList:

---

[8] Indication of management regime for specified CodeLists has not been included in this prototype model

[9] Met Office is currently developing a pilot Registry implementation for WMO TDCF Code-tables



Thus each element relating to a FlagList shall be stereotyped «FlagList». For example: Table B Class 20 [0 20 021] 'Type of precipitation'

«FlagList» B20-021_PrecipitationTypeValues

[Bit] no.	
1	Precipitation-unknown type
2	Liquid precipitation not freezing
3	Liquid freezing precipitation
4	Drizzle
5	Rain
6	Solid precipitation
7	Snow
8	Snow grains
9	Snow pellets
10	Ice pellets
11	Ice crystals
12	Diamond dust
13	Small hail
14	Hail
15	Glaze
16	Rime
17	Soft rime
18	Hard rime
19	Clear ice
20	Wet snow
21	Hoar frost
22	Dew
23	White dew
24-29	Reserved

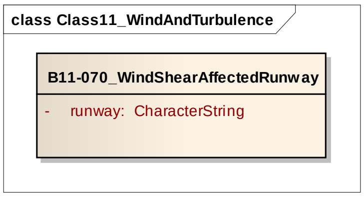
[All] 30	Missing value

An XML implementation of this might be:

```
<B20-021_PrecipitationTypeValues flagList="http://{hostname}/{path}/Code+Flag/B_20_021">
        <element flagListValue"http://{hostname}/{path}/Code+Flag/B_20_021/5">Rain</element>
        <element flagListValue"http://{hostname}/{path}/Code+Flag/B_20_021/20">Wet snow</element>
</B20-021_PrecipitationTypeValues>
```

Other Table B types

Occasionally, a BUFR/CREX Code is specified with 'unit' as "CCITT IA5" (Character String: International Alphabet No. 5)implying that free-form text is permitted; e.g. Code [0 11 070] 'Designator of the runway affected by wind shear (including ALL)'. Thus the derived Class 'B11-070_WindShearAffectedRunway' would assert the following properties:



These Classes are anomalous; it would seem more appropriate to relate the attribute to some specific type of entity. However, the BUFR model permits the use of CharacterString for this entity. As such, Classes that follow this pattern are dealt with on a case-by-case basis.

Inferring a Logical Data Model from BUFR Table D Sequences

In contradiction of the original proposal from WMO TT-AvXML [M1][10] the Class structure shall be inferred by looking at how 'Generalised Coordinates' (BUFR/CREX Table B Classes 1 -7) and 'Significance Qualifiers' (BUFR/CREX Table B Class 8) are used.

For information, the Generalised Coordinate Classes from BUFR/CREX Table B are listed below:

- Class 01: Identification
- Class 02: Instrumentation
- Class 04: Location (time)
- Class 05: Location (horizontal - 1)
- Class 06: Location (horizontal - 2)
- Class 07: Location (vertical)

[Note: Class 03 is 'Reserved']

Class 01 'Identification' is used to identify Observing Stations and other platforms (aircraft, ships), geographic (WMO) Regions and specific named Features (e.g. aerodromes (ICAO code), storms, aerosol sources etc.). Additional information about the motion of a mobile platform is also included here - although this is probably for convenience.

Class 02 'Instrumentation' provides information about the instrumentation used in observing event. A brief analysis suggests a strong parallel with the type of information expected in OM_Process (ISO 19156

[10] e.g. where BUFR Table D Sequences were identified as the primary discriminating factors for identifying Classes

Observation and Measurements). As such, changes in instrumentation may provide an appropriate discriminator to identify the presence of discrete Observation instances.

Class 04 'Location (time)' specifies the component elements required to specify a 'time-position' (time instant), time-range, increment (including specific reference time periods for establishing statistical measurements e,g, max/min, accumulation etc. These entities (of collections thereof) should map to ISO 19108 'Geographic information - Temporal Schema'.

Class 05 'Location (horizontal - 1)' and Class 06 'Location (horizontal - 2)' specify coordinate positions (Latitude / Longitude), various directions (e.g. bearing or azimuth, solar azimuth) and information about satellite scans. These may relate to the position of the observing platform or the position of the phenomenon that is being observed (these will be different for remote-sensing and synthetic observation/simulation activities).

Class 07 'Location (vertical)' specifies information about the general vertical positions/increments, vertical position of an Observation Platform, the relative vertical position of a Sensor with respect to local reference position (e.g. ground/water surface) and the vertical position of the observed phenomenon (e.g. 0 07 012 'Grid point altitude'). Vertical position may be expressed in height, geopotential , (water) pressure, flight level etc.

It is harder to generalise the impact of 'Significance Qualifiers' (Class 08) as the entities almost exclusively relate to Code- and Flag-tables. However, they seem to imply some context to the observation. Thus the presence of a Significance Qualifier _MAY_ be a discriminator to identify the presence of a discrete observation.

Repetition of a Generalised Coordinate or Significance Qualifier indicates the 'closure' of the previous group. In some cases the Table D Sequences include a Generalised Coordinate or Significance Qualifier annotated to indicate that it is set to 'missing' - thus explicitly closing the group.

Use of Replication Factors will also be used as discriminators to determine how to modularise the Template.

To assess this method, we must ignore the Sequence structure and fully expand the BUFR Template (see below). The Table D Sequence numbers are retained for reference as this will be useful in comparing with the previous approach. These rows are in BLUE text but should be ignored when looking at the information content of the message (line numbers are not given for Sequence elements). Date, Time and (horizontal) Position sequences have not been 'unpacked' as these are clear candidates for harmonisation with Classes from the ISO/TC 211 reference models. Generalised Coordinates (Class 01 - Class 07) are identified in PURPLE. Class 08 Significance Qualifiers (Class 08) are identified in DARK RED. Yellow highlight indicates where new Classes have been identified.

Notes:

- New Table D Sequences are identified with values starting from the 301 – these cannot be confused with _REAL_ BUFR Sequence numbers as these are limited to 0-255. Generally, new Sequences are created within Class 07 ‘Surface reports – land’. Exceptionally, a new ForecastTime group is specified in Table D Class 01 ‘Location and Identification’.
- Generally, replication is defined via the cardinality of associations or attributes within the UML model.
- Where Table D Sequences specify replication for multiple Table B elements (e.g. Table D Sequence [3 07 046] ‘Visbility’ repeats elements [0 05 021] ‘Bearing or azimuth’ and [0 20 059] ‘Minimum visibility’ as a pair), these must be refactored into separate Classes – this is the only method available to ensure that ALL of the elements in the replication group are repeated consistently. We note that frequently, such Classes definitions include a Generalised Coordinate or Significance Qualifier, suggesting that they would have been identified as new Classes anyhow. The replication limits of the group are specified via the cardinality of the UML association.
- Table D Sequences that specify replication of a single Table B element (e.g. [3 07 014] ‘Weather intensity and phenomena’) are ‘collapsed’ such that Table B element is specified directly as an attribute with the appropriate multiplicity.
- The occurrence of any Generalised Coordinate or Significance Qualifier is used to identify where a new Class must be created.

- Generalised Coordinates or Significance Qualifiers shall apply to all elements in the Class within which it is defined plus all elements of child-Classes. The Generalised Coordinate or Significance Qualifier comes first and applies to all simple descriptors in the class as well as all child classes. This forces cleaner modifier specifications (e.g. barometer height and QNH).

Gil Ross (Met Office) additionally notes:

- Many of the new groupings are listed as Table D Class 07 'Surface reports – land'. These probably ought to be refactored as Table D Class 2 'Common sequences for surface data'.
- As for the de-normalised descriptors windSpeed_ms-1 windSpeed_knots etc, we should develop a fully normalised descriptor of windSpeed, perhaps with a constraint of class 8 which defines units which has aliases as the denormalised Table B descriptors

#	Code	Repl. group	ExistingElementName_en	Proposed shortname / Class Name	C	Target class
	307051		Full METAR/SPECI	**D07-051_METAR-SPECI**		
	307045		Main part of METAR/SPECI data			
1	001063		ICAO location indicator CCCC	icaoLocationIndicator	1	B01-063_ICAOLocationIdentifier
2	008079		Aviation product status (routine, special, corrected, not available) METAR SPECI COR	productStatus	1	B08-079_AviationProductStatus
3	002001		Type of station(AUTO)	stationType	1	B02-001_StationType
4	301011		Year, month, day YY	date	1	D01-011_Date
5	301012		Hour, minute GGgg	time	1	D01-012_Time
6	301023		Latitude-longitude (coarse accuracy)	position	1	D01-023_Position_coarse
7	007030		Height of station ground above mean sea level	stationHeight	1	B07-030_StationHeight
~~8~~	~~007031~~	-	~~Height of barometer above mean sea level~~	~~barometerHeight~~	~~1~~	~~B07-031_BarometerHeight~~
+	307303			product	1	D07-303_METAR-SPECI_Product
+	307311			windGroup	1	D07-311_WindGroup
9	007032		Height of sensor above local ground = 10 m (if the actual value is not available)	windSensorHeight	1	B07-032_SensorHeight
10	011001		Wind direction ddd	windDirection	1	B11-001_WindDirection
11	011016		Extreme counterclockwise wind direction of a variable wind dndndn	variableWindDirectionCCW	1	B11-016_VariableWindDirectionCCW
12	011017		Extreme clockwise wind direction of a variable	variableWindDirectionCW	1	B11-017_VariableWindDirectionC

Comment [J1]: Lines 1-7 provide context for the entire Record: ICAO location indicator, product status, date, time, position & height of station. These are retained in the top-level Class D07-051_METAR-SPECI and apply to ALL elements in the Template

Comment [J2]: Refactored so that it is adjacent to 'Altimeter setting (QNH)' in line 26

Comment [J3]: New Class created following specification of 'global' Generalised Coordinates and Significance Qualifiers

Comment [J4]: New Class created to presence of Generalised Coordinate [0 07 032] in line 9

			wind dxdxdx			W
+	307307			windSpeedGroup	1	D07-307_WindSpeedGroup
13	008054		Qualifier for wind speed or wind gusts P	windSpeedQualifier	1	B08-054_WindSpeedOrGustQualifier
14	011083		Wind speed (km/h) ff	windSpeed_kmh-1	1	B11-083_WindSpeed_kmh-1
15	011084		Wind speed (knots) ff	windSpeed_knots	1	B11-084_WindSpeed_knots
16	011002		Wind speed (m/s) ff	windSpeed_ms-1	1	B11-002_WindSpeed_ms-1
close Class D07-307_WindSpeedGroup						
+	307308			windGustGroup	1	D07-308_WindGustGroup
17	008054		Qualifier for wind speed or wind gusts P	windGustQualifier	1	B08-054_WindSpeedOrGustQualifier
18	011085		Maximum wind speed (gusts) (km/h) fmfm	maxWindSpeed_kmh-1	1	B11-085_MaxWindSpeed_kmh-1
19	011086		Maximum wind speed (gusts) (knots) fmfm	maxWindSpeed_knots	1	B11-086_MaxWindSpeed_knots
20	011041		Maximum wind speed (gusts) (m/s) fmfm	maxWindSpeed_ms-1	1	B11-041_MaxWindSpeed_ms-1
21	008054		Qualifier for wind speed or wind gusts = missing (to cancel the previous value)			
close Class D07-308_WindGustGroup						
close Class D07-311_WindGroup						
+	307310			spotTemperature	1	D07-310_SpotTemperature
22	007032		Height of sensor above local ground = 2 m (if the actual value is not available)	temperatureSensorHeight *thermometerHeight?*	1	B07-032_SensorHeight
23	012023		Temperature (Celsius) TT	temperature	1	B12-023_Temperature
24	012024		Dew point (Celsius) TdTd	dewpoint	1	B12-024_Dewpoint
25	007032		Height of sensor above local ground = missing (to cancel the previous value)			
close Class D07-310_SpotTemperature						
+	307309			stationPressure	1	D07-309_QNH
8	007031		Height of barometer above mean sea level	barometerHeight	1	B07-031_BarometerHeight
26	010052		Altimeter setting (QNH) QPHPHPHPH	qnh	1	B10-052_QNH
close Class D07-309_QNH						

Comment [J5]: New Class created due to presence of Significance Qualifier [0 08 054] in line 13

Comment [J6]: Class closed due to reset of Significance Qualifier [0 08 054] in line 17

Comment [J7]: New Class created due to presence of Significance Qualifier [0 08 054] in line 17

Comment [J8]: Class closed due to reset of Significance Qualifier [0 08 054] in line 21

Comment [J9]: Class closed due to reset of Generalised Coordinate [0 07 032] in line 22

Comment [J10]: New Class created to presence of Generalised Coordinate [0 07 032] in line 22

Comment [J11]: Class closed due to reset of Generalised Coordinate [0 07 032] in line 25

Comment [J12]: New Class created to presence of Generalised Coordinate [0 07 031] in line 25A (moved from line 8)

Comment [J13]: Moved from line-8; Significance Qualifier applies only to 'Altimeter setting (QNH)'

Comment [J14]: Class closed because we infer that QNH is the only element to which 'barometerHeight' is applicable

27	020009		General weather indicator TAF/METAR CAVOK	weatherIndicator	1	B20_009_WeatherIndicator
	307046		Visibility VVVV or VVVVNDV VNVNVVVNDV			
28	020060		Prevailing visibility VVVV or VVVVNDV	prevailingVisibility	1	B20-060_PrevailingVisibility
29	102000	-	Delayed replication of two descriptors			
30	031001	[0..2]	Number of replication (up to 2)			
+	307302			minVisibility	0..2	D07-302_DirectedMinVisibility
31	005021	1	Bearing or azimuth (direction of minimum visibility observed) Dv	direction	1	B05-021_Bearing
32	020059	2	Minimum visibility VNVNVNVN	minVisibility	1	B20-059_MinVisibility
close Class D07-302_DirectedMinVisibility						
	307013		Runway visual range RDRDR/VRVRVRVR			
33	106000	-	Delayed replication of 6 descriptors			
34	031001	[0..4]	Number of replication (up to 4)			
+	307013			runwayVisualRange	0..4	D07-013_RunwayVisualRangeGroup
35	001064	1	Runway designator	runway	1	B01-064_RunwayDesignator
+	307306			Rvr	2	D07-306_RunwayVisualRange
36	008014	2	Qualification for runway visual range	qualifier	1	B08-014_RVRQualifier
37	020061	3	Runway visual range	Range	1	B20-061_RunwayVisualRange
close Class D07-306_RunwayVisualRange						
~~+~~	~~307XXX~~			~~rvr1~~	~~1~~	~~D07-XXX_RunwayVisualRange~~
~~38~~	~~008014~~	4	~~Qualification for runway visual range~~	~~qualifier~~	~~1~~	~~B08-014_RVRQualifier~~
~~39~~	~~020061~~	5	~~Runway visual range~~	~~range~~	~~1~~	~~B20-061_RunwayVisualRange~~
40	020018	6	Tendency of runway visual range	rvrTendency	1	B20-018_RVRTendency
close Class D07-013_RunwayVisualRangeGroup						
	307014		Weather intensity and phenomena w┤w┤		0..3	~~D07-014_WeatherIntensityAndPhenomena~~

Comment [J15]: New Class created to presence of Generalised Coordinate [0 05 021] in line 31

Comment [J16]: Multiplicity set based on replication note

Comment [J17]: Class closed as replication note (line 29) specified 2-descriptors to repeat only

Comment [J18]: New Class created to presence of Generalised Coordinate [0 01 064] in line 35

Comment [J19]: New Class created to normalise explicit repetition in BUFR Table D Sequence ([qualifier]+[rvr] pair) – here we use multiplicity [2]

Comment [J20]: Class closed as replication note (line 33) specified 6-descriptors to repeat only

						B20-019_PresentWeather
41	101000	-	Delayed replication of 1 descriptor			
42	031001	[0..3]	Number of replication (up to 3)			
43	020019	1	Significant present weather	presentSignificantWx	0..3	B20-019_PresentWeather
	307047		Clouds NsNsNshshshs			
44	105000	-	Delayed replication of 5 descriptor			
45	031001	[?]	Number of replications			
+	307301			clouds	0..?	D07-301_CloudGroup
46	008002	1	Vertical significance	verticalSignificance	1	B08-002_VerticalSignificance
47	020011	2	Cloud amount NsNsNs	cloudAmount	1	B20-011_CloudAmount
48	020012	3	Cloud type CC	cloudType	1	B20-012_CloudType
49	020013	4	Height of base of cloud (m) hshshs	cloudBaseHeight_m	1	B20-013_CloudBaseHeight_m
50	020092	5	Height of base of cloud (feet) hshshs	cloudBaseHeight_ft	1	B20-092_CloudBaseHeight_ft
close Class D07-301_CloudGroup						
51	020002		Vertical visibility (m) VVhshshs	verticalVisbility_m	1	B20-002_VerticalVisibility_m
52	020091		Vertical visibility (feet) VVhshshs	verticalVisbility_ft	1	B20-091_VerticalVisibility_ft
	307016		Recent weather phenomena REw┬┤w┬┤			
53	101000	-	Delayed replication of 1 descriptor			
54	031001	[0..3]	Number of replication (up to 3)			
55	020020	1	Significant recent weather phenomena	recentSignificantWx	0..3	B20-020_RecentWeather
	307017		Runway shear WS RDRDR			
56	101000	-	Delayed replication of 1 descriptor			
57	031001	[?]	Number of replication			
58	011070	1	Runway designator of the runway affected by wind shear (including ALL) *(this is an observation of which runways are affected by wind-shear – the 'featureOfInterest'*	runwayShear	0..?	B11-070_WindShearAffectedRunway

Comment [J21]: No new Class created – multiplicity 'promoted' to attribute

Comment [J22]: New Class created to presence of Significance Qualifier [0 08 002] in line 46

Comment [J23]: Class closed as replication note (line 44) specified 5-descriptors to repeat only

Comment [J24]: No new Class created – multiplicity 'promoted' to attribute

Comment [J25]: No new Class created – multiplicity 'promoted' to attribute

			remains the aerodrome)			
	307049		Sea conditions WTsTs/SS┬┤			
59	102000	-	Delayed replication of 2 descriptors			
60	031000	[0..1]	Short delayed replication factor (0 or 1)			
+	307049			seaConditions	0..1	D07-049_SeaConditions
61	022043	1	Sea/water temperature TsTs	waterTemperature	1	B22-043_WaterTemperature
62	022021	2	Height of waves S┬┤	waveHeight	1	B22-021_WaveHeight
close Class D07-049_SeaConditions						
	307050		Runway state RDRDR/ERCReReRBRBR			
63	101000	-	Delayed replication of 1 descriptor			
64	031000	[0..1]	Short delayed replication factor (0 or 1)			
65	020085	1	General condition of runway SNOCLO	runwayState	0..1	B20-085_RunwayCondition
66	102000	-	Delayed replication of 2 descriptors			
67	031001	[?]	Number of replications			
+	307305			runwayCondition	0..?	D07-305_RunwayConditionSummary
68	001064	1	Runway designator DRDR	runway	1	B01-064_RunwayDesignator
69	020085	2	General condition of runway CLRD//	condition	1	B20-085_RunwayCondition
close Class D07-305_RunwayConditionSummary						
70	105000	-	Delayed replication of 5 descriptors			
71	031001	[?]	Number of replications			
+	307304			runwayConditionDetail	0..?	D07-304_RunwayConditionDetails
72	001064	1	Runway designator DRDR	runway	1	B01-064_RunwayDesignator
73	020086	2	Runway deposits ER	deposits	1	B20-086_RunwayDeposits
74	020087	3	Runway contamination CR	contamination	1	B20-087_RunwayContamination
75	020088	4	Depth of runway deposits eReR	depthOfDeposit	1	B20-088_RunwayDepositDepth
76	020089	5	Runway friction	frictionCoefficient	1	B20-

Comment [J26]: Existing Sequence can be reused

Comment [J27]: No new Class created – multiplicity 'promoted' to attribute

Comment [J28]: New Class created to presence of Generalised Coordinate [0 01 064] in line 68

Comment [J29]: Class closed as replication note (line 66) specified 2-descriptors to repeat only

Comment [J30]: New Class created to presence of Generalised Coordinate [0 01 064] in line 72

			coefficient BRBR			089_RunwayFrictionCoefficient
close Class D07-304_RunwayConditionDetails						
77	101000	-	Delayed replication of 1 descriptor			
78	310010	[0..3]	Replication count (0 to 3 normally)			
+	307048			trendForecast	0..3	D07-048_TrendForecast
	307048		Trend type forecast			
79	008016		Change qualifier for trend type forecast TTTTT NOSIG	changeIndicator	1	B08-016_TrendForecastChangeQualifier
80	102000	-	Delayed replication of 2 descriptors			
81	031001	[0..2]	Number of replications (0, 1 or 2)			
+	301301			timeGroup	0..2	D01-301_ForecastTimeOfChange
82	008017	1	Qualifier for time of forecast change TT	qualifier	1	B08-017_TimeQualifierForForecastedChange
83	301012	2	Time of change GGgg	time	1	D01-012_Time
close Class D01-301_ForecastTimeOfChange						
+	307312			windGroup	1	D07-312_ForecastWindGroup
84	007032		Height of sensor above local ground = 10 m (if the actual value is not available)	windSensorHeight	1	B07-032_SensorHeight
85	011001		Wind direction ddd	windDirection	1	B11-001_WindDirection
+	307307			windSpeedGroup	1	D07-307_WindSpeedGroup
86	008054		Qualifier for wind speed or wind gusts P	windSpeedQualifier	1	B08-054_WindSpeedOrGustQualifier
87	011083		Wind speed (km/h) ff	windSpeed_kmh-1	1	B11-083_WindSpeed_kmh-1
88	011084		Wind speed (knots) ff	windSpeed_knots	1	B11-084_WindSpeed_knots
89	011002		Wind speed (m/s) ff	windSpeed_ms-1	1	B11-002_WindSpeed_ms-1
close Class D07-307_WindSpeedGroup						
+	307308			windGustGroup	1	D07-308_WindGustGroup
90	008054		Qualifier for wind speed or wind gusts P	windGustQualifier	1	B08-054_WindSpeedOrGustQualifier
91	011085		Maximum wind speed	maxWindSpeed_kmh-1	1	B11-

Comment [J31]: Class closed as replication note (line 70) specified 5-descriptors to repeat only

Comment [J32]: Re-use existing Sequence. Also note fit with presence of Significance Qualifier [0 08 016] in line 79

Comment [J33]: New Class created to presence of Significance Qualifier [0 08 017] in line 82

Comment [J34]: Significance Qualifier [0 08 017] relates to a Code-table with the following entries:
- [0] FM (from)
- [1] TL (until)
- [2] AT (at)

The Manual on Codes (WMO306_Vol_I.1 para 15.14.3) states that the timeGroup shall be used to indicate the beginning (FM) or the end (TL) of a forecast change, or the time (AT) at which specific forecast condition(s) is (are) expected.

Comment [J35]: Class closed as replication note (line 80) specified 2-descriptors to repeat only

Comment [J36]: New Class created to presence of Generalised Coordinate [0 07 032] in line 84

Comment [J37]: Re-use new Sequence defined earlier

Comment [J38]: Re-use new Sequence defined earlier

			(gusts) (km/h) fmfm			085_MaxWindSpeed_kmh-1
92	011086		Maximum wind speed (gusts) (knots) fmfm	maxWindSpeed_knots	1	B11-085_MaxWindSpeed_knots
93	011041		Maximum wind speed (gusts) (m/s) fmfm	maxWindSpeed_ms-1	1	B11-041_MaxWindSpeed_ms-1
94	008054		Qualifier for wind speed or wind gusts = missing (to cancel the previous value)			
close Class D07-308_WindGustGroup						
95	007032		Height of sensor above local ground = missing (to cancel the previous value)			
close Class D07-312_ForecastWindGroup						
96	020009		General weather indicator CAVOK NSW NSC	weatherIndicator	1	B20_009_WeatherIndicator
97	101000	-	Delayed replication of 1 descriptor			
98	031000	[0..1]	Short delayed replication count (0 or 1)			
99	020060	1	Prevailing visibility VVVV	prevailingVisbility	0..1	B20-060_PrevailingVisibility
	307014		Weather intensity and phenomena w┬┤w┬┤			
100	101000	-	Delayed replication of 1 descriptor			
101	031001	[0..3]	Number of replication (up to 3)			
102	020019	1	Significant present weather	presentSignificantWx	0..3	B20-019_PresentWeather
	307047		METAR/SPECI/TAF clouds NsNsNshshshs	clouds	1	D07-047_Clouds
103	105000	-	Delayed replication of 5 descriptor			
104	031001	[?]	Number of replications			
+	307047			clouds	0..?	D07-047_Clouds
105	008002	1	Vertical significance	verticalSignificance	1	B08-002_VerticalSignificance
106	020011	2	Cloud amount NsNsNs	cloudAmount	1	B20-011_CloudAmount
107	020012	3	Cloud type CC	cloudType	1	B20-012_CloudType
108	020013	4	Height of base of cloud (m) hshshs	cloudBaseHeight_m	1	B20-013_CloudBaseHeight_m
109	020092	5	Height of base of cloud (feet) hshshs	cloudBaseHeight_ft	1	B20-092_CloudBaseHeight_ft

Comment [J39]: Class closed due to re-set of Generalised Coordinate [0 07 032] in line 95

Comment [J40]: No new Class created – multiplicity 'promoted' to attribute

Comment [J41]: No new Class created – multiplicity 'promoted' to attribute

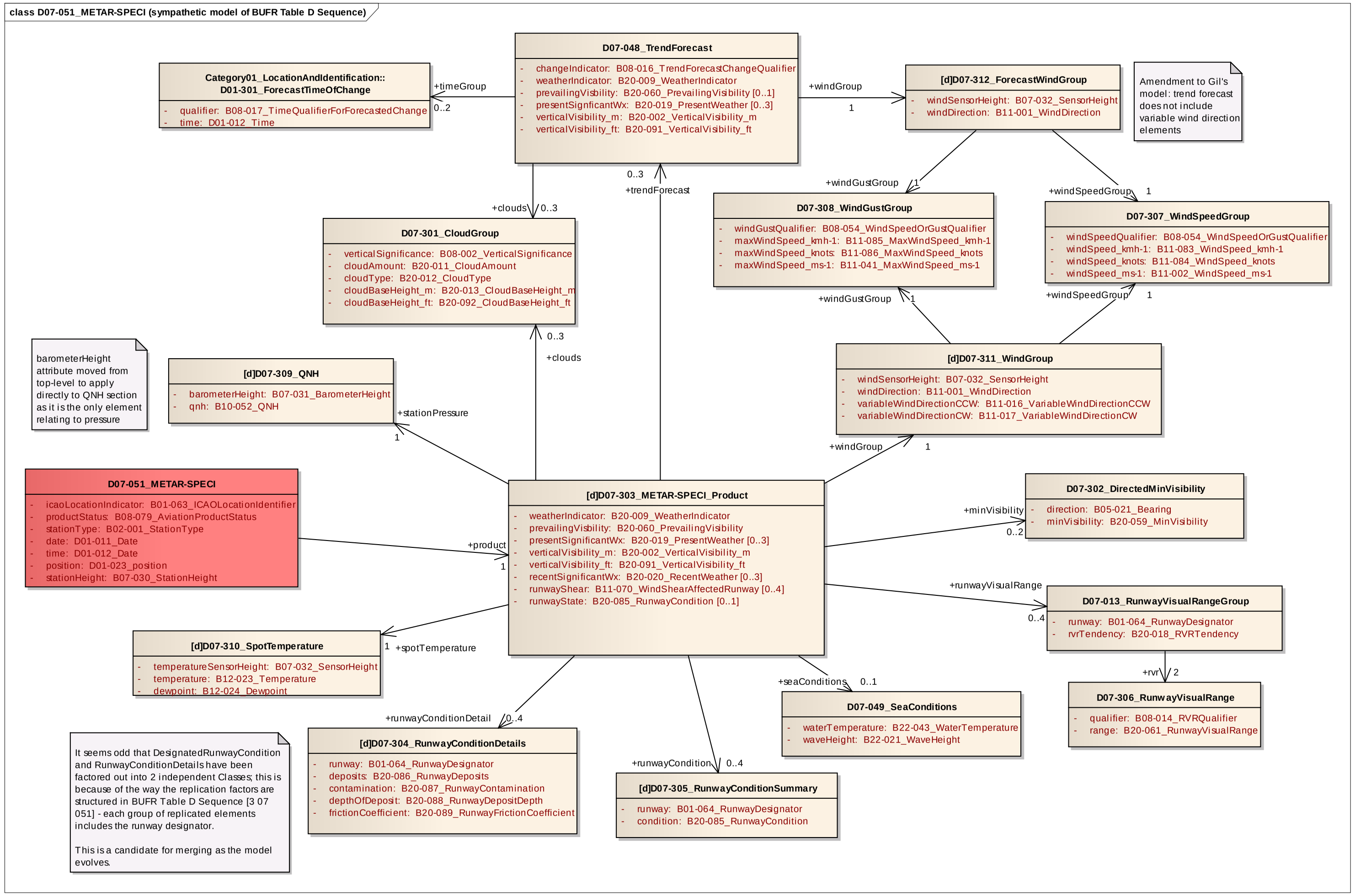
Comment [J42]: Re-use new Sequence defined earlier

close Class D07-047_Clouds						
110	020002		Vertical visibility (m) VVhshshs	verticalVisbility_m	1	B20-002_VerticalVisibility_m
111	020091		Vertical visibility (feet) VVhshshs	verticalVisbility_ft	1	B20-091_VerticalVisibility_ft
close Class D07-048_TrendForecast						
close Class D07-303_METAR-SPECI_Product						
close Class D07-051_METAR_SPECI						

Notes:

[Lines 68 & 72] It seems odd that DesignatedRunwayCondition and RunwayConditionDetails have been factored out into 2 independent Classes; this is because of the way the replication factors are structured in BUFR Table D Sequence [3 07 051] - each group of replicated elements includes the runway designator. This is a candidate for merging as the model evolves - the group of detailed conditions could become a sub-group of the summary condition. I also note that RunwayVisualRangeGroup is also factored out into a separate Class; this too is a candidate for merging.

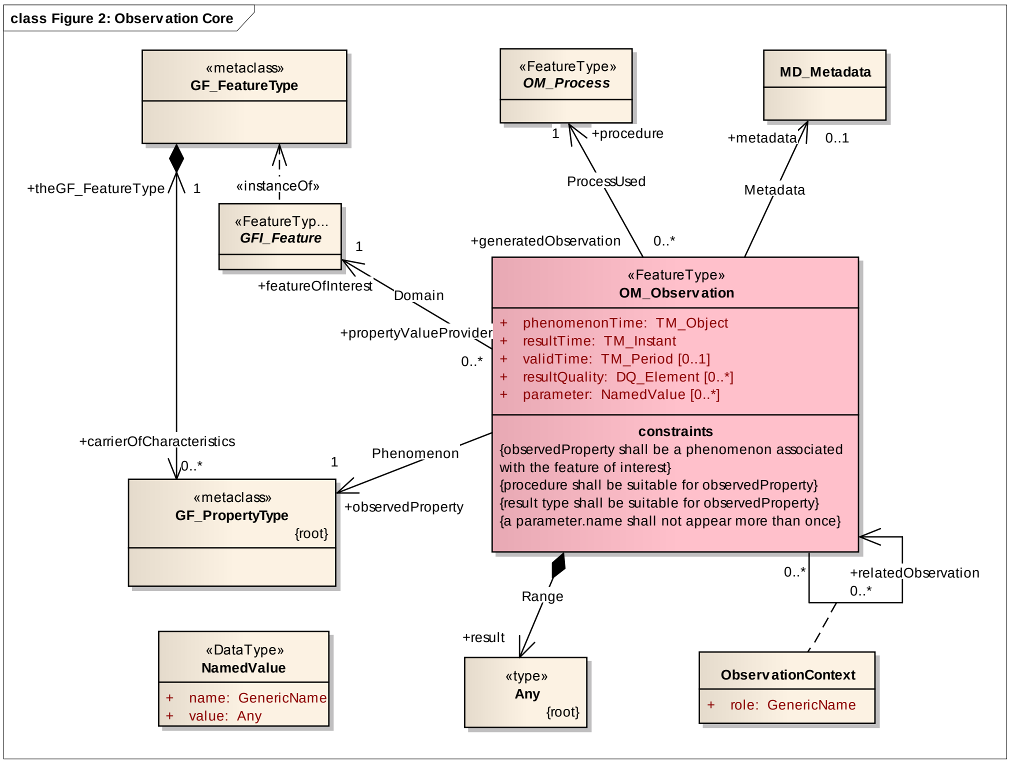
The METAR/SPECI model is shown on the next page. *Class-names prefixed with [d] indicate that the Class is deprecated in the 'ISO/TC 211 harmonised' METAR/SPECI model.*



class D07-051_METAR-SPECI (sympathetic model of BUFR Table D Sequence)
D07-048_TrendForecast
- changeIndicator: B08-016_TrendForecastChangeQualifier
- weatherIndicator: B20-009_WeatherIndicator
- prevailingVisibility: B20-060_PrevailingVisibility [0..1]
- presentSignficantWx: B20-019_PresentWeather [0..3]
- verticalVisibility_m: B20-002_VerticalVisibility_m
- verticalVisibility_ft: B20-091_VerticalVisibility_ft
Category01_LocationAndIdentification::
D01-301_ForecastTimeOfChange
- qualifier: B08-017_TimeQualifierForForecastedChange
- time: D01-012_Time
+timeGroup
0..2
+windGroup
1
[d]D07-312_ForecastWindGroup
- windSensorHeight: B07-032_SensorHeight
- windDirection: B11-001_WindDirection
Amendment to Gil's model: trend forecast does not include variable wind direction elements
+clouds
0..3
0..3
+trendForecast
+windGustGroup
1
+windSpeedGroup
1
D07-301_CloudGroup
- verticalSignificance: B08-002_VerticalSignificance
- cloudAmount: B20-011_CloudAmount
- cloudType: B20-012_CloudType
- cloudBaseHeight_m: B20-013_CloudBaseHeight_m
- cloudBaseHeight_ft: B20-092_CloudBaseHeight_ft
D07-308_WindGustGroup
- windGustQualifier: B08-054_WindSpeedOrGustQualifier
- maxWindSpeed_kmh-1: B11-085_MaxWindSpeed_kmh-1
- maxWindSpeed_knots: B11-086_MaxWindSpeed_knots
- maxWindSpeed_ms-1: B11-041_MaxWindSpeed_ms-1
D07-307_WindSpeedGroup
- windSpeedQualifier: B08-054_WindSpeedOrGustQualifier
- windSpeed_kmh-1: B11-083_WindSpeed_kmh-1
- windSpeed_knots: B11-084_WindSpeed_knots
- windSpeed_ms-1: B11-002_WindSpeed_ms-1
+windGustGroup
1
+windSpeedGroup
1
0..3
+clouds
barometerHeight attribute moved from top-level to apply directly to QNH section as it is the only element relating to pressure
[d]D07-309_QNH
- barometerHeight: B07-031_BarometerHeight
- qnh: B10-052_QNH
+stationPressure
1
[d]D07-311_WindGroup
- windSensorHeight: B07-032_SensorHeight
- windDirection: B11-001_WindDirection
- variableWindDirectionCCW: B11-016_VariableWindDirectionCCW
- variableWindDirectionCW: B11-017_VariableWindDirectionCW
+windGroup
1
D07-051_METAR-SPECI
- icaoLocationIndicator: B01-063_ICAOLocationIdentifier
- productStatus: B08-079_AviationProductStatus
- stationType: B02-001_StationType
- date: D01-011_Date
- time: D01-012_Date
- position: D01-023_position
- stationHeight: B07-030_StationHeight
+product
1
[d]D07-303_METAR-SPECI_Product
- weatherIndicator: B20-009_WeatherIndicator
- prevailingVisibility: B20-060_PrevailingVisibility
- presentSignificantWx: B20-019_PresentWeather [0..3]
- verticalVisibility_m: B20-002_VerticalVisibility_m
- verticalVisibility_ft: B20-091_VerticalVisibility_ft
- recentSignificantWx: B20-020_RecentWeather [0..3]
- runwayShear: B11-070_WindShearAffectedRunway [0..4]
- runwayState: B20-085_RunwayCondition [0..1]
+minVisibility
0..2
D07-302_DirectedMinVisibility
- direction: B05-021_Bearing
- minVisibility: B20-059_MinVisibility
+runwayVisualRange
0..4
D07-013_RunwayVisualRangeGroup
- runway: B01-064_RunwayDesignator
- rvrTendency: B20-018_RVRTendency
+rvr
2
D07-306_RunwayVisualRange
- qualifier: B08-014_RVRQualifier
- range: B20-061_RunwayVisualRange
[d]D07-310_SpotTemperature
- temperatureSensorHeight: B07-032_SensorHeight
- temperature: B12-023_Temperature
- dewpoint: B12-024_Dewpoint
1
+spotTemperature
+seaConditions
0..1
D07-049_SeaConditions
- waterTemperature: B22-043_WaterTemperature
- waveHeight: B22-021_WaveHeight
+runwayConditionDetail
0..4
[d]D07-304_RunwayConditionDetails
- runway: B01-064_RunwayDesignator
- deposits: B20-086_RunwayDeposits
- contamination: B20-087_RunwayContamination
- depthOfDeposit: B20-088_RunwayDepositDepth
- frictionCoefficient: B20-089_RunwayFrictionCoefficient
+runwayCondition
0..4
[d]D07-305_RunwayConditionSummary
- runway: B01-064_RunwayDesignator
- condition: B20-085_RunwayCondition
It seems odd that DesignatedRunwayCondition and RunwayConditionDetails have been factored out into 2 independent Classes; this is because of the way the replication factors are structured in BUFR Table D Sequence [3 07 051] - each group of replicated elements includes the runway designator.
This is a candidate for merging as the model evolves.

Harmonisation with standard models

ISO 19156 Observation and Measurements



ISO 19156 'Geographic information - Observations and Measurements' provides a candidate baseline for the WMO data model; this is compliant with WXXM2.0 as planned.

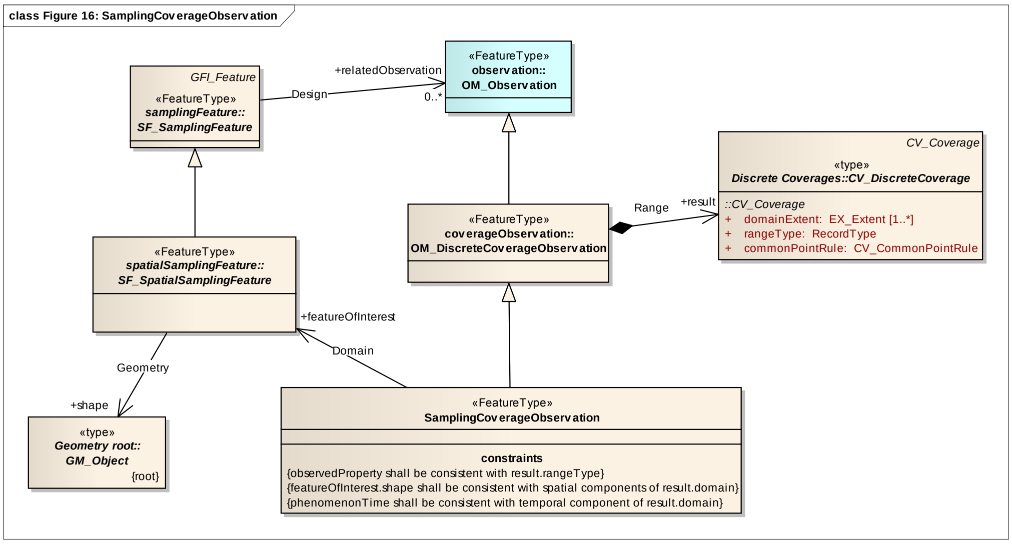
We note that the semantics of OM_Observation state: 'an estimate of the value of some property of some Thing using some specified Process'. The *process* may be an instrument/sensor (directly) measuring some physical parameter or a numerical simulation predicting future values. Thus OM_Observation may be used to represent both observations and forecasts.

WXXM groups several 'Observations' into a Report. The Report conforms to a Data Product Specification – which can be described via a Logical Data Model / Application Schema.

Climate Science Modelling Language (CSML)

During TT-AvXML Meeting 1, the team discussed how the earth-science communities appear to be converging toward a data model that is expressed in CSML3; Unidata's Common Data Model (CDM) which underpins netCDF, Climate & Forecast conventions etc. We also noted that CSML3 lacks adequate documentation to be referenced as a definitive standard. CSML3 has been issued as an OGC best practice document: OGC 11-021. The INSPIRE Data Specification drafting teams / thematic working groups for Atmospheric Conditions / Meteorological Features (AC/MF) and Oceanographic Features (OF) have included CSML3 'patterns' in their data-specifications; i.e. the INSPIRE data models follow the best-practices defined in CSML3 but do not explicitly reference the CSML3 standard. The INSPIRE Annex 2 & 3 Comments Resolution Workshop (Ispra, 5-7 Dec 2011) recommended that the CSML3 model become part of the Generic Conceptual Model that underpins all INSPIRE Data Specifications.

Alignment of the WMO data model with CSML3 best-practices should ensure interoperability with the earth-science community (particularly climate science) and European INSPIRE Data Specifications.



CSML3 is predicated on the (informative) SamplingCoverageObservation (above) defined in ISO 19156; all the CSML3 Observation classes (Feature Types) derive from this SamplingCoverageObservation Feature Type. The SamplingCoverageObservation applies a set of important constraints to the OM_Observation model:

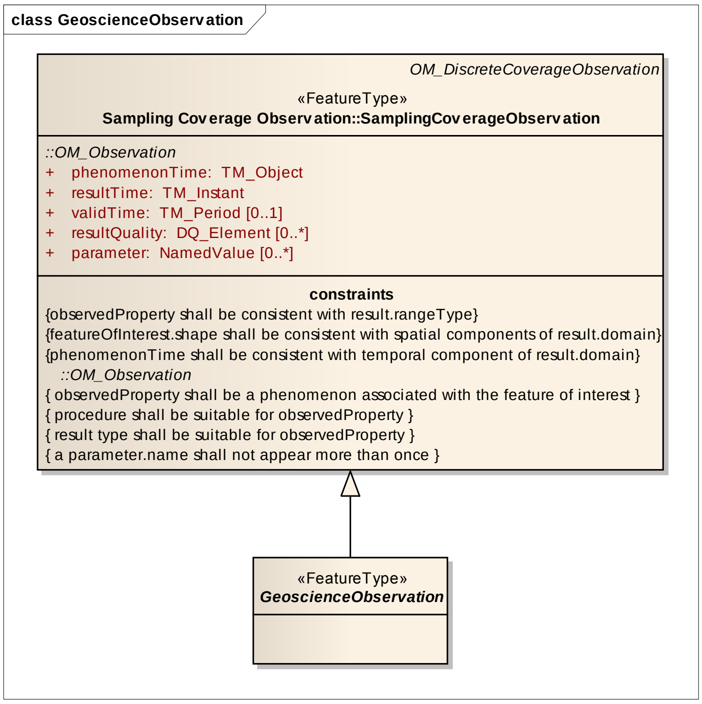
- the *featureOfInterest* property shall refer to an instance of SF_SamplingFeature (nominally one of SF_SampingPoint, SF_SamplingCurve, SF_SamplingSurface or SF_SampingSolid),
- the *result* property shall refer to an instance of CV_DiscreteCoverage – or subclass thereof,
- the shape of the *featureOfInterest* property shall be consistent with spatial elements of result domain,
- the *phenomenonTime* property shall be consistent with temporal elements of result domain, and
- the *observedProperty* property shall be consistent with the range type of the result.

ASSERTION: WMO Logical Data Model shall comply with the constraints inherited from SamplingCoverageObservation constraints.

Each of the CSML3 Observation classes further specialises SampingCoverageObservation based on the spatio-temporal 'shape' of the dataset domain.

Due to documentation issues, the CSML3 model has been incorporated into the WMO Logical Data Model within the 'GeoscienceObservations' package. The *pattern* used is identical to CSML3 – only the names of Classes have changed.

CSMLObservation (GeoscienceObservation) is the base abstract Class from which all other CSML3 classes derive.



CSML3 declares two top-level abstract classes:

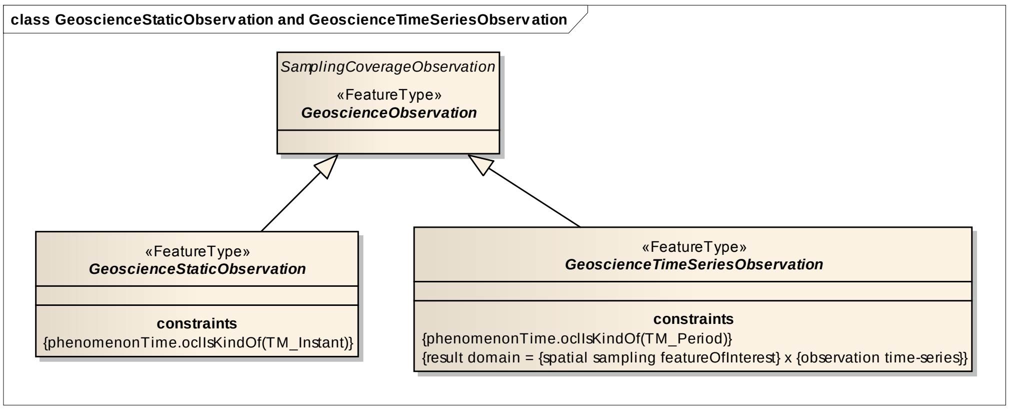
- *CSMLStaticObservation* (GeoscienceStaticObservation) - representing ‘static’ Observations (i.e. those where the observation result covers an instant in time. **Constraint**:

  `phenomenonTime.oclIsKindOf(TM_Instant)` : The observation phenomenon time is a time instant (TM_Instant from ISO 19108)

- CSMLTimeSeriesObservation (GeoscienceTimeSeriesObservation) – representing Observations with repetitive sampling that generates a time-series. **Constraint**:

  `phenomenonTime.oclIsKindOf(TM_Period)` : The observation phenomenon time is a time period (TM_Period from ISO 19108) representing the temporal extent of the repeat observation time-series

  `result domain = {spatial sampling feature-of-interest} × {observation time-series}` : The spatiotemporal domain of the discrete coverage result consists, mathematically, of the product of the sampling points (located on a spatial sampling feature) and a time-series of repeated observation times



The following classes from CSML3 appear relevant to the WMO Logical Data Model in the short- to medium-term:

- *Point* (PointObservation)
- *PointSeries* (PointSeriesObservation)
- *Grid* (GridObservation)
- *GridSeries* (GridSeriesObservation)

CSML3 defines other specialised observation classes that are likely to be of use in future iterations of the WMO Logical Data Model:

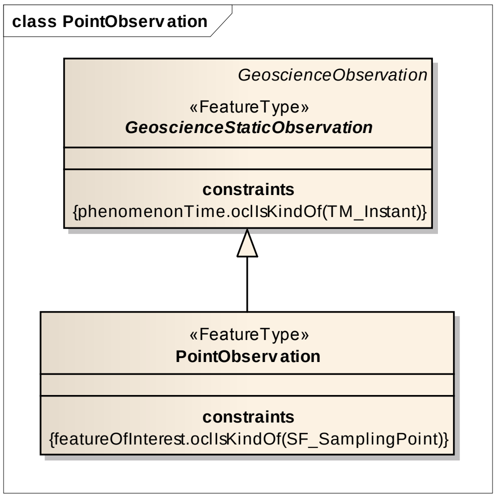
- *Profile*
- *ProfileSeries*
- *Trajectory*
- *Section*
- *Swath*
- *ScanningRadar*

Please refer to document OGC 11-021 for further information.

PointObservation

The Point observation is a CSML Static Observation that represents a measurement of a property at a single point in time and space. The *result* property of a Point observation shall be a discrete coverage having a domain of a single point (CV_DiscretePointCoverage from ISO 19123). **Contraint**:

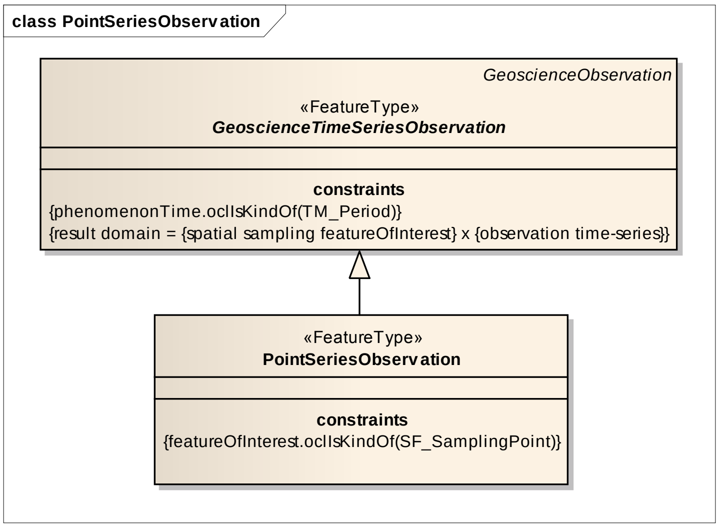
```
featureOfInterest.oclIsKindOf(SF_SamplingPoint)
```



PointSeriesObservation

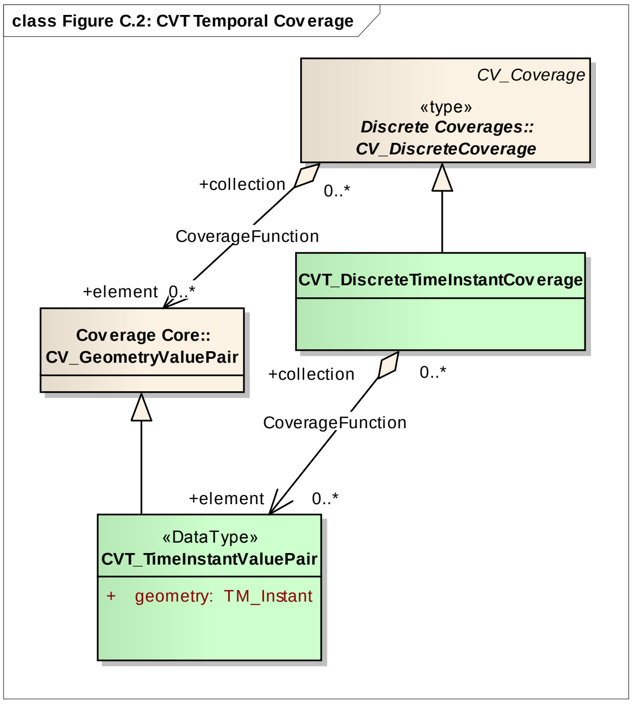
The Point Series Observation is a CSML Time Series Observation that represents a repeated sampling of a property at a fixed location in space. The result of a Point Series observation shall be a discrete coverage with domain elements at a single fixed location but with temporal values corresponding to the series of sampling times (CV_DiscretePointCoverage from ISO 19123). **Constraint**:

```
featureOfInterest.oclIsKindOf(SF_SamplingPoint)
```



Note: the CSML3 documentation states that the domain of the result Coverage shall be expressed within a spatio-temporal CRS (Coordinate Reference System); e.g. each domain value shall have up to 3 spatial dimensions and one time dimension – (x, y, z, T). Given that the domain is a fixed spatial location, this introduces significant redundancy with multiple repetition of the (x, y, z) coordinates. Furthermore, we note that the location information is also provided explicitly within the SF_SamplingPoint instance associated to the Observation via the featureOfInterest property. Thus a simple temporal reference system could be used to express the domain of the time series; e.g. 0, 1, 2, 3, 4, … where the CRS describes hours since 2011-12-13T12:00Z.

An alternative mechanism is the use of CVT_DiscreteTimeInstantCoverage, defined within ISO 19156:



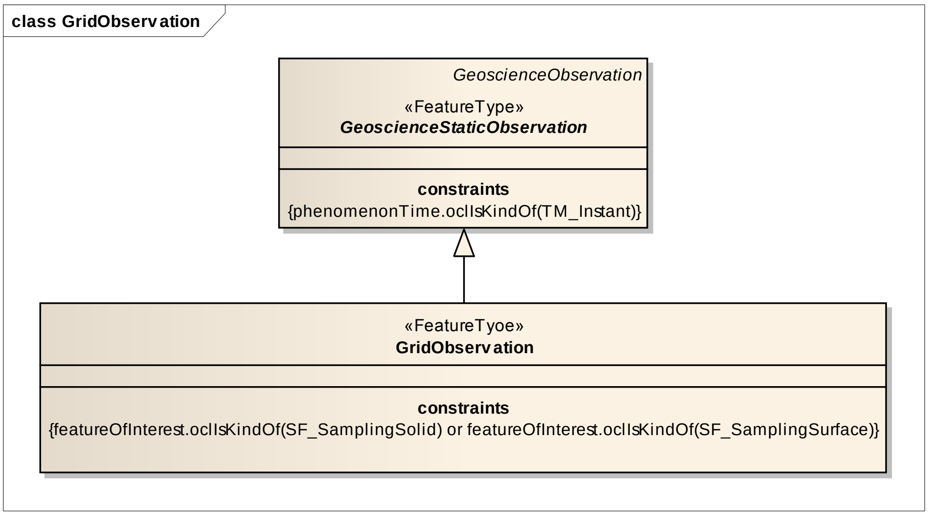
CVT_DiscreteTimeInstantCoverage specialises the typical CV_DiscreteCoverage by specifying that the 'geometry' of the domain value may be expressed as a TM_Instant. Thus the domain of the coverage would be expressed using ISO 8601 tokens: 2011-12-13T12:00Z, 2011-12-13T13:00Z, 2011-12-13T14:00Z, 2011-12-13T15:00Z, 2011-12-13T12:00Z etc.

Strictly speaking, one cannot validate the constraint [`result domain = {spatial sampling feature-of-interest} × {observation time-series}`] where the coverage domain has no spatial aspect. Thus one may consider the methods expressed here to be invalid; i.e. invalid constraint = FAIL. However, it is equally valid to state that the invalid constraint has UNKNOWN result – which could be PASS or FAIL. Clearly we are *inferring* that the domain of the result is bound to the specific point location – thus the constraint can be inferred to be true.

GridObservation

The Grid Observation is a CSML Static Observation representing a gridded field (2 or 3 dimensions) at a single time instant. The result of a Grid observation is a discrete coverage where the domain consists of a two- or three-dimensional grid of points, all having the same time instant temporal component. Typically the result will be an instance of CV_DiscreteGridPointCoverage, where the domain points are determined according to some scheme or function; simple grid specification, rectified grid or referenceable grid (ISO 19123 Geographic information – Coverages refers). **Constraint**:

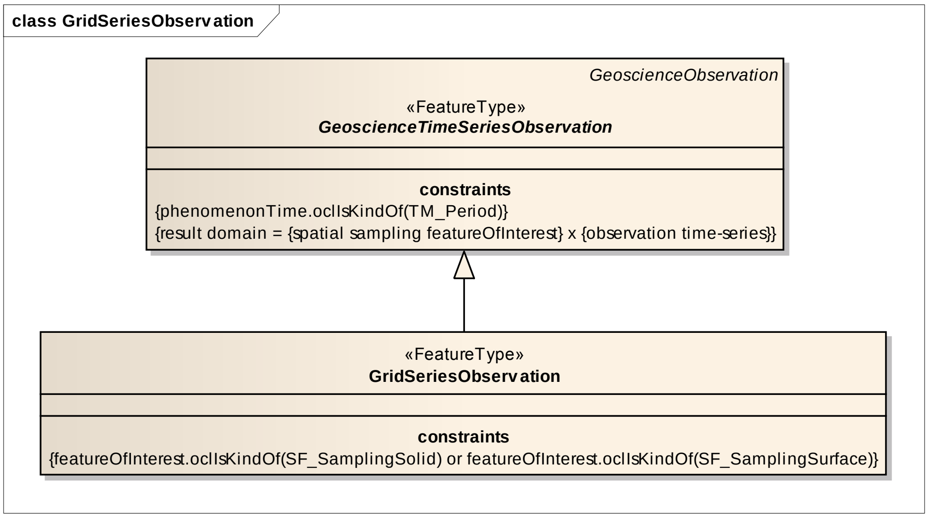
```
featureOfInterest.oclIsKindOf(SF_SamplingSolid) or
featureOfInterest.oclIsKindOf(SF_SamplingSurface)
```



GridSeriesObservation

The Grid Series Observation is a CSML Time Series Observation representing an evolving gridded field (2 or 3 dimensions) at a succession of time instants. The result of a Grid observation is a discrete coverage with a compound spatio-temporal CRS where the domain consists of a series of two- or three-dimensional grid of points, each at a successive time instant. Typically the result will be an instance of CV_DiscreteGridPointCoverage. **Constraint**:

```
featureOfInterest.oclIsKindOf(SF_SamplingSolid) or
featureOfInterest.oclIsKindOf(SF_SamplingSurface)
```



Note: the Grid Series observation can be used to express measurements sampled at a geographically distributed set of locations over a period of time – assuming that the property and the time instants at which data is sampled are uniform at each location. Two methods are possible:

(1) use of CV_DiscretePointCoverage, wherein each (x, y, *z,* T) coordinate for each point in the domain is provided explicitly – noting the repetitive duplication of the (x, y, *z*) coordinates for each time instant at which the data is sampled (e.g. the points are at fixed spatial locations and do not vary with time).

(2) use ReferenceableGridCoverage [GML Change Request OGC 07-112r3 refers], wherein the 3- or 4-dimensional coverage (up to 3 spatial dimensions plus time) are flattened to a 2-dimensional grid; index-space plus time (i, t) where each point in index-space maps to the spatial sampling location (e.g. the location of a weather station within an observing network.

Option (2) provides a much more compact mechanism to express the domain points; this is the mechanism that is typically used within netCDF files. However, the use of ReferenceableGridCoverage adds significant complexity to encode/decode the dataset.

Recommendation: where possible, a complex observation with a sampling regime that spans multiple geographic locations and multiple time-instants shall be exchange as a collection of simple observations; either PointSeries observations (one per geographic location) or MultiPoint Observation (one per time-instant, see below).

Beyond CSML3

During the Testing phase of INSPIRE Annex 2 & Annex 3 Data Specifications, Met Office identified that CSML3 lacked the ability to express data sampled from a spatially distributed locations; e.g. an observing network or collection of sites for which localised forecasts are provided.

In the WMO context, a requirement arises from US proposals regarding the exchange of surface observations. Currently, a surface observation report contains a co-temporal set of measurements taken (logically) at the same location (e.g. SYNOP or METAR). The US proposal suggests that a surface observation report contains co-temporal set of measurements of same type (i.e. Temperature measurements) collected from several sites. Different collections of sites may be used for different measurement types (depending on where particular sensors are deployed).

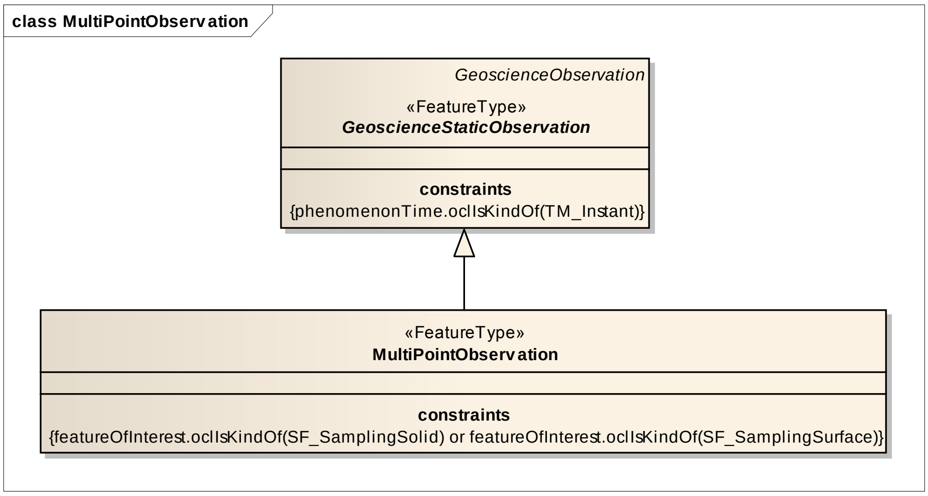
This requirement is typical of the desire to exchange data sampled from a set of geographically distributed locations at a single time-instant.

To remedy this deficiency in CSML3, a new Observation type is proposed: MultiPoint. The topology of the MultiPoint observation is orthogonal to the PointSeries observation; e.g. MultiPoint describes spatially distributed sampling at a time-instant whilst PointSeries describes a time-series of samples at a specific fixed location.

MultiPointObservation

The MultiPoint observation is a CSML Static Observation that represents a co-temporal measurement of a property at a collection of spatial locations at a specific time instant. The *result* property of a MultiPoint observation shall be a discrete coverage having a domain that describes the locations at which the measurement occurred (CV_DiscretePointCoverage from ISO 19123). **Contraint**:

```
featureOfInterest.oclIsKindOf(SF_SamplingSurface)
```



Note: constraint assumes that all locations can be fitted to a surface; e.g. ground surface or a pressure level. Alternatively, permitting featureOfInterest to be of type SF_SamplingSolid would allow the locations to be randomly distributed in 3-dimensions.

Note: it is possible to consider the MultiPoint observation as a sub-type of the Grid observation; e.g. using a ReferenceableGridCoverage to collapse the (x, y, *z*) coordinates of the set of spatial points onto a 1-dimensional grid in 'index-space'. However, use of ReferenceableGridCoverage in this context is significantly more complex than CV_DiscretePointCoverage. Grid observation does not explicitly constrain the result to be an instance of CV_DiscreteGridPointCoverage – only the shape of the result domain is constrained. However, the name of the class implies use of a DiscreteGridPointCoverage. The MultiPoint observation provides clearer semantics and explicitly permits use of a simple CV_DiscretePointCoverage for the result.

Alternative coverage domain geometries

It is also conceivable that the domain objects in a discrete coverage may be of higher dimensionality than a simple point; e.g. linear or bounded by a polygon.

For example: a SIGMET may be defined via a line (e.g. EGPX SCOTTISH FIR SEV TURB OBS **S OF LINE N6000** BLW FL060 STNR INTSF=) or within a polygon.

> Note: it is always possible to derive a bounding polygon for a SIGMET based on the intersection of the line projected in the specified direction. However, it is important that a SIGMET can be communicated verbally; 'S OF LINE N6000' is much easier to communicate verbally than a bounding polygon!

Where the domain objects are lines (Curve) or areas bounded by polygon (Surface), CV_DiscreteCurveCoverage and CV_DiscreteSurfaceCoverage should be used.

Following the pattern developed for MultiPoint observation, where the sampled data is associated with Curves or Surfaces at a single time-instant, MultiCurve and MultiPoint observations should be used.

MultiCurveObservation

The MultiCurve observation is a CSML Static Observation that represents a co-temporal measurement of a property at a collection of spatial locations at a specific time instant. The *result* property of a MultiCurve observation shall be a discrete coverage having a domain that describes the set of curves at which the measurement occurred (CV_DiscreteCurveCoverage from ISO 19123). **Contraint**:

```
featureOfInterest.oclIsKindOf(SF_SamplingSurface) or
featureOfInterest.oclIsKindOf(SF_SamplingSolid)
```

MultiSurfaceObservation

The MultiSurface observation is a CSML Static Observation that represents a co-temporal measurement of a property at a collection of spatial locations at a specific time instant. The *result* property of a Point observation shall be a discrete coverage having a domain that describes the set of curves at which the measurement occurred (CV_DiscreteCurveCoverage from ISO 19123). **Contraint**:

```
featureOfInterest.oclIsKindOf(SF_SamplingSurface) or
featureOfInterest.oclIsKindOf(SF_SamplingSolid)
```

Modelling 'time'

The Observation model has several explicit time attributes:

- phenomenonTime
- resultTime
- validTime

These terms all have well defined semantics. The difference is best explained with an example:

> *An ice-core specimen is extracted from a glacier. The time of extraction is the phenomenonTime. Later, the ice-core specimen is subjected to tests from which results are published. The time(s) at which the result is (results are) generated is the resultTime.*

In case of forecasts or other numerical simulations, the result time shall be different from the phenomenon time. Often, a forecast simulation will have a phenomenonTime in the future[11].

Modelling the Observation result

The WMO Logical Data Model aims to derive from SamplingCoverageObservation. This enforces the constraint that the Observation result shall be a sub-Class of {Abstract} CV_DiscreteCoverage (and the Observation featureOfInterest shall be a sub-Class of {Abstract} SF_SpatialSamplingFeature).

From analysis of the BUFR data model, we infer that the format is designed to express data that is _INTERLEAVED_ with the geometry (e.g. as each geometry point is defined, the data values associated with that point are also expressed).

In comparison with the Coverage schema (from ISO 19123) - this equates to the _DISCRETE POINT COVERAGE_ model (CV_DiscretePointCoverage).

For the time being, we shall limit[12] our interest in CSML3 Observation types to:

- *CSML3:Point* (PointObservation)
- *CSML3:PointSeries* (PointSeriesObservation)
- *MultiPoint* (MultiPointObservation)

We note that GRIB is designed to express _GRIDDED_ data and so *may* be used to encode other types of CSML3 Observation: 'CSML3:Grid' and 'CSML3:GridSeries'[13].

Degenerate coverages

At first glance, it would appear excessive introduce the complexity of a (discrete) coverage model where the data is sampled only at a single domain object; e.g. a SYNOP contains observations from a single weather station whilst a SIGMET will typically use a single bounding polygon.

---

[11] Although the model is called 'Observations and Measurements' the semantics still apply to (weather) forecasts. Whilst challenging for our community to disassociate the term 'observation' from the (physical) measurement of the atmospheric state, an OM_Observation with a Procedure that is a complex numerical simulation is clearly able to estimate values of some Property (e.g. a temperature field) in the future.

[12] more complex geometries for SIGMETs will be discussed in a future release of this document

[13] Pending discussion on GRIB3

This scenario aligns with the CSML3:Point Observation (PointObservation).

However, the overhead introduced from the use of discrete coverages to define the observation result are arguably small, and yet enable flexibility to provide more sophisticated information at later dates as the expectation of data publishers and consumers evolves. For example, the US proposal for exchanging Surface Observations suggests that a surface observation report contains co-temporal set of measurements of same type (i.e. Temperature measurements) collected from several sites, whilst a TS SIGMET (thunderstorm) may contain multiple zones each at a different flight level – or perhaps at a different intensity.

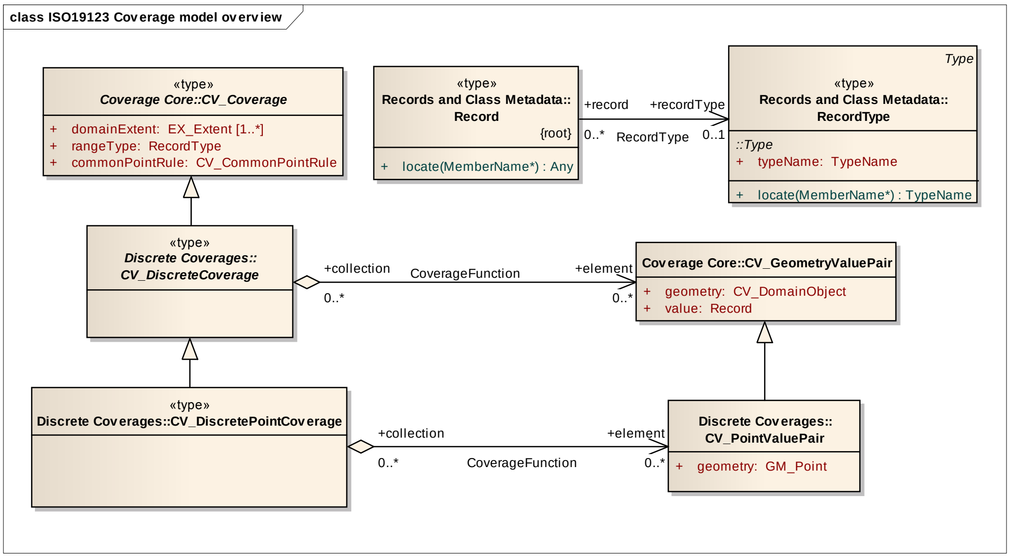
We also note that BUFR is frequently used to express data about observed phenomena from a set of geometry points within a single message. The WMO Logical Data Model must be able to support this scenario.

This scenario aligns with the MultiPoint Observation (MultiPointObservation).

Finally, we may wish to describe a time-series of measurements taken at a single geometry.

This scenario aligns with the CSML3:PointSeries Observation (PointSeriesObservation).

ISO 19123 Geographic information - Schema for coverage geometry and functions



A coverage is used to model the variation of some property within a domain of interest. Thus a coverage comprises of a set of domain elements (the geometries or time-instants), a set of range elements (the data values), a common point rule procedure to map from the domain to the range and a description of the types of data found in the range (the rangeType). Coverages of various types are specified in ISO 19123 - including CV_DiscreteGridPointCoverage where the domain elements are defined implicitly using a grid-point definition (e.g. as might be described by Section 3 of a GRIB2 message).

As identified above, our interest relates to CV_DiscretePointCoverage.

The CV_DiscretePointCoverage comprises of a number of elements; two of these are important for our discussion here:

- rangeType:RecordType
- element:CV_GeometryValuePair

rangeType

ISO 19123 specifies 'rangeType':

> *"The attribute rangeType:RecordType shall describe the range of the coverage. The data type RecordType is defined in ISO/TS 19103. It consists of a list of attribute name/data type pairs. A simple list is the most common form of rangeType, but RecordType can be used recursively to describe more complex structures. The rangeType for a specific coverage shall be specified in an application schema."*

The ISO 19123 Coverage model asserts that RecordType is a 'list of properties'.

Within GML/XML RecordType is typically implemented using a SWE:DataRecord which could be populated using information from the BUFR Tables. In this situation, DataRecord/field/Quantity/definition shall be a URI that refers to the appropriate BUFR Table B definition*. Also note that specifying the unit of measure is mandatory for a SWE:Quantity.

> Note: within a BUFR Sequence[14], the rangeType attribute may be implemented using Table B Class 00 code [0 00 030] 'Descriptor defining sequence' to refer to the appropriate BUFR Table D Sequence number that defines the appropriate set of properties[15].

For example:

```
<rangeType>
   <swe:DataRecord definition="http://{hostname}/{path}/BUFR/D/07/404">
      <swe:lable>Temperature observation record</swe:lable>
      <swe:field name="temperature">
         <swe:Quantity definition="http://{hostname}/{path}/BUFR/B/12/023">
            <swe:uom code="Cel"/>
         </swe:Quantity>
      </swe:field>
      <swe:field name="dewpoint">
         <swe:Quantity definition="http://{hostname}/{path}/BUFR/B/12/024">
            <swe:uom code="Cel"/>
         </swe:Quantity>
      </swe:field>
   </swe:DataRecord>
</rangeType>
```

Note: SWE:Quantity may also carry a 'value' - this is not required here.

Again, we note that BUFR Table B definitions will need to be published as web-accessible definitions - each with a unique identifier (URI).

The equivalent description in BUFR is significantly smaller as it simply references a Sequence from BUFR Table D using code [0 00 030] 'Descriptor defining sequence':

Code	**Value**

[14] We aspire to be able to encode the WMO Logical Data Model in TDCF in addition to XML/GML

[15] we note that new Table D Sequences will be required to describe some of the rangeSets - e.g. D07-307_WindSpeedGroup which contains four attributes: windSpeedQualifier [0 08 054], windSpeed_kmh-1 [0 11 083], windSpeed_knots [0 11 084] and windSpeed_ms-1 [0 11 002]

[0 00 030]	307404

ASIDE: Qualified properties

Complex Properties such as [0 12 011]'Maximum temperature, at height and over period specified' rely on the preceding definition of vertical coordinate and duration.

One option may be to define new Table D Sequences to express the Qualifiers adjacent to the Measure - and then reference such Sequences from the rangeSet. For example:

```
3 YY XXX 'Maximum temperature, at height and over period specified'
        0 07 032 'height of sensor above local ground (m)'
        0 04 074 'period (hrs)'
        0 12 011 'Maximum temperature, at height and over period specified'
```

However, this would imply that each time the Record was instantiated one would need to provide the height and period values. Normally, one would anticipate that ALL members of a coverage result would conform to the same constraints - e.g. all instances of maximum temperature would be from a sensor at the same height, calculated over the same period. It is clear that this information is part of the processing methodology - and thus should be defined within the Process Class (see below). Additionally, this information will also be defined within the ObservedProperty (see below).

In this situation, we must rely on those previous definitions to inform the interpretation of the measured value.

Thus a code such as [0 12 011] 'Maximum temperature, at height and over period specified' would appear directly within a Record - relying on previous definitions of the height and period.

Furthermore, in XML one is able reference elements previously defined within the scope of a file. A _QUALIFIED_ property such as [0 12 011] 'Maximum temperature, at height and over period specified' will be fully described as part of the Observation observedProperty (see details later of the ObservableProperty model). Thus an explicit local reference (within the XML file) can be made to identify the *qualifiers* that are used for this specific measure.

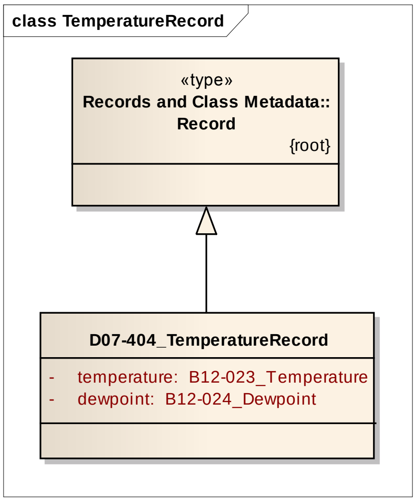
GeometryValuePair

ISO 19123 specifies the 'CV_GeometryValuePair' Class:

"The class CV_GeometryValuePair describes an element of a set that defines the relationships of a discrete coverage. Each member of this class consists of two parts: a domain object from the spatiotemporal domain of the coverage to which it belongs and a record of feature attribute values from the range of the coverage to which it belongs."

In the case of CV_DiscretePointCoverage, the geometry attribute is specialised to Class type 'GM_Point'. The attribute 'value' is identified as type 'Record' - this shall be an instance of the RecordType defined previously.

Thus each BUFR Table D Sequence that is used to group properties together shall be modelled as sub-Classes of Record. The names of such Classes should terminate in 'Record' for clarity. For example, Class D07-404_TemperatureRecord:



Classes B12-023_Temperature and B12-024_Dewpoint have attributes 'value' and 'uom'[16].

When encoded in BUFR, the code values used within the Sequence (e.g. [0 12 023] and [0 12 024]) shall provide information about how to encode each value (e.g. Scale, Reference value and Data width).

GML/XML serialisation of the TemperatureRecord:

```xml
<value>
  <D07-404_TemperatureRecord>
    <temperature>
      <!-- unit of measure from WMO Common-table C-6: [350] 'celsius' -->
      <B12-023_Temperature uom="http://{hostname}/{path}/Common/C-6/350">
        27
      </B12-023_Temperature>
    </temperature>
    <dewpoint>
      <!-- unit of measure from WMO Common-table C-6: [350] 'celsius' -->
      <B12-024_Dewpoint uom="http://{hostname}/{path}/Common/C-6/350">
        10
      </B12-024_Dewpoint>
    </dewpoint>
  </D07-404_TemperatureRecord>
</value>
```

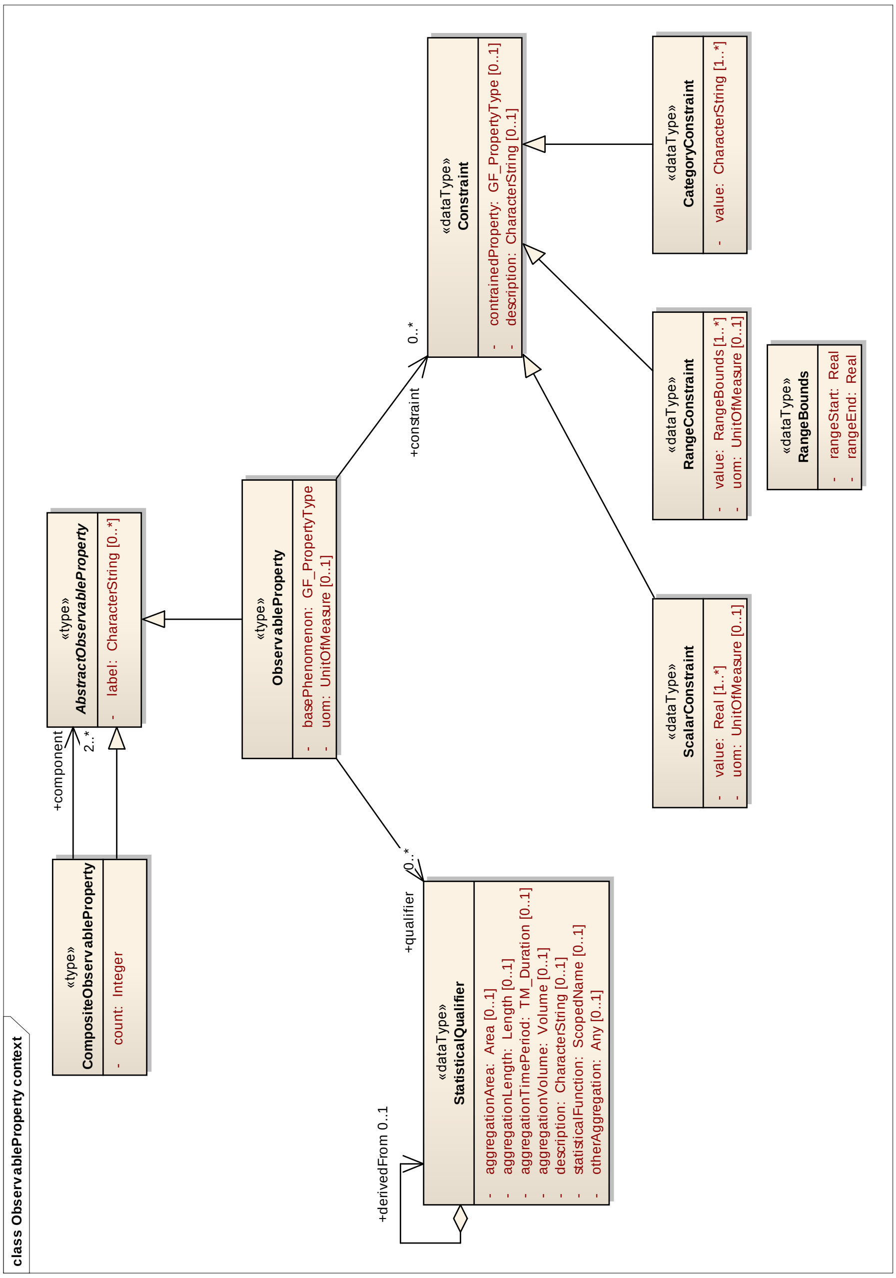
Note: 'uom' has been encoded as an XML attribute - this seems appropriate. It may be more appropriate for units of measure to reference authoritative definitions via URI rather than use a simple text string.

To enforce the use of a particular Record type within the CV_PointValuePair (and ensure that the rangeType refers to the correct target) constraints shall be defined in an enclosing class.

[16] constrained to Celsius

Modelling the Observation observedProperty

The 'Phenomenon' Role from ISO19156 Observations and Measurements relates the Observation to the Phenomenon measured or observed. Only a single Phenomenon may be expressed per Observation (cardinality is specified as [1] in OM_Observation). To counter this problem, Dominic Lowe (CEDA, UK) has been leading the development of the 'ObservableProperty' model within the OGC SWE-Common Standards Working Group. The model is not yet endorsed, but shows promise:



The two significant uses of this model are:

- To group multiple 'simple' Observable Properties into a single 'CompositeObservableProperty' instance that can be referenced by OM_Observation via the observedProperty associations.
- To explicitly define constraints on 'base phenomena' such as types of statistic operator and scalar constraints (e.g. basePhenomena 'temperature' with statistical qualifier 'maximum' and scalar constraint on property 'period', value '12' and unit of measure 'hours').

There will be some duplication between the rangeType definition in the Coverage result and the observedProperty.

Where only a single Property is Observed, one can directly reference an instance of ObservedProperty. For example:

```
<observedProperty>
   <ObservableProperty>
      <label>
         <CharacterString>Altimeter setting - QNH (Pa)</CharacterString>
      </label>
      <basePhenomenon xlink:href="http://{hostname}/{path}/BUFR/B/10/052"/>
      <!-- unit of measure from WMO Common-table C-6: [032] 'pascal' -->
      <uom xlink:href="http://{hostname}/{path}/Common/C-6/032"/>
   </ObservableProperty>
</observedProperty>
```

Attribute 'uom' is optional - and not strictly required in the example above because the 'basePhenomenon' defintion referenced (BUFR Table B Class 10 code [0 10 052] already declares the units as 'Pa'. Unit of measure is referenced from WMO Common Code-table C-6.

While no appropriate BUFR Table D Sequences exist for the ObservableProperty model, these should be possible to define. The specification should include BUFR Table B Class 0 code [0 00 030] 'Descriptor defining sequence'[17] to specify the 'basePhenomenon' association; e.g.

Code	Shortname	Value	Description
[0 00 030]	basePhenomenon	010052	Altimeter setting - QNH

Multiple Properties can be grouped together into a CompositeObservableProperty:

```
<observedProperty>
   <CompositeObservableProperty>
      <label>Aviation temperature measure group</label>
      <count>2</count>
      <component>
         <ObservableProperty>
            <label>
               <CharacterString>Temperature (C)</CharacterString>
            </label>
            <basePhenomenon xlink:href="http://{hostname}/{path}/BUFR/B/12/023"/>
            <!-- unit of measure from WMO Common-table C-6: [350] 'celsius' -->
            <uom xlink:href="http://{hostname}/{path}/Common/C-6/350"/>
         </ObservableProperty>
      </component>
      <component>
         <ObservableProperty>
            <label>
               <CharacterString>Dew-point temperature (C)</CharacterString>
```

[17] or variant thereof - we pointing to descriptor from Table B, not a Sequence from Table D

```
                </label>
                <basePhenomenon xlink:href="http://{hostname}/{path}/BUFR-CREX/B/12/024"/>
                <!-- unit of measure from WMO Common-table C-6: [350] 'celsius' -->
                <uom xlink:href="http://{hostname}/{path}/Common/C-6/350"/>
              </ObservableProperty>
            </component>
          </CompositeObservableProperty>
        </observedProperty>
```

A more complex example of a single ObservableProperty would be the _QUALIFIED_ property [0 12 011] 'Maximum temperature, at height and over period specified' – in this example, qualified at height = 2m over period 12hrs:

```
<ObservableProperty>
  <label>
    <CharacterString>Maximum temperature, at height and over period specified</CharacterString>
  </label>
  <basePhenomenon xlink:href="http://{hostname}/{path}/BUFR-CREX/B/12/011"/>
  <uom>K</uom>
  <qualifier>
    <StatisticalQualifier>
      <!-- BUFR-CREX Code-table [0 08 023] 'First order statistics': [2] 'Maximum value' -->
      <statisticalFunction xlink:href="http://{hostname}/{path}/BUFRCREX/Code+Flag/08.023/2"/>
    </StatisticalQualifier>
  </qualifier>
  <constraint>
    <ScalarConstraint>
      <constrainedProperty xlink:href="http://{hostname}/{path}/BUFR/B/07/032"/>
      <description>
        <CharacterString>Height above local ground (m)</CharacterString>
      </description>
      <!-- unit of measure from WMO Common-table C-6: [001] 'metre' -->
      <uom xlink:href="http://{hostname}/{path}/Common/C-6/001"/>
      <value>2.0</value>
    </ScalarConstraint>
  </constraint>
  <constraint>
    <ScalarConstraint>
      <constrainedProperty xlink:href="http://{hostname}/{path}/BUFR/B/04/074"/>
      <description>
        <CharacterString>Period (hrs)</CharacterString>
      </description>
      <!-- unit of measure from WMO Common-table C-6: [131] 'hour' -->
      <uom xlink:href="http://{hostname}/{path}/Common/C-6/131"/>
      <value>12</value>
    </ScalarConstraint>
  </constraint>
</ObservableProperty>
```

Note: Measures such as [0 12 004] 'Air temperature at 2m' are similar to qualified properties. However, they have already been explicitly defined within BUFR - so we _MAY_ use them directly as the basePhenomena - although it is warping the semantics somewhat! For example:

```
<ObservableProperty>
  <label>
    <CharacterString>Air Temperature at 2m</CharacterString>
  </label>
  <basePhenomenon xlink:href="http://{hostname}/{path}/BUFR/B/12/004"/>
  <!-- unit of measure from WMO Common-table C-6: [005] 'kelvin' -->
  <uom xlink:href="http://{hostname}/{path}/Common/C-6/005"/>
</ObservableProperty>
```

This is equivalent to:

```
<ObservableProperty>
  <label>
    <CharacterString>Air Temperature at 2m</CharacterString>
  </label>
  <basePhenomenon xlink:href="http://{hostname}/{path}/BUFR/B/12/001"/>
  <!-- unit of measure from WMO Common-table C-6: [032] 'kelvin' -->
  <uom xlink:href="http://{hostname}/{path}/Common/C-6/005"/>
  <constraint>
```

```
            <ScalarConstraint>
                <constrainedProperty xlink:href="http://{hostname}/{path}/BUFR/B/07/032"/>
                <description>
                    <CharacterString>Height above local ground (m)</CharacterString>
                </description>
                <!-- unit of measure from WMO Common-table C-6: [001] 'metre' -->
                <uom xlink:href="http://{hostname}/{path}/Common/C-6/001"/>
                <value>2.0</value>
            </ScalarConstraint>
        </constraint>
    </ObservableProperty>
```

Note: basePhenomenon refers to [0 12 001] 'Temperature / air temperature' instead of [0 12 004] as in the previous example.

Note: the GML/XML implementation of OM_Observation specifies association 'observedProperty' as 'byReference' only - meaning that the association can only be specified via an xlink reference (URI) to some target resource. This will need to be address in the resulting GML/XML schema - most easily by overriding this setting in the specialised GeoscienceObservation Classes.

Modelling the Observation process

The 'Procedure' Role from ISO19156 Observations and Measurements relates the Observation to the Procedure used to execute that observation. ISO 19156 Observations and Measurements defines the Class 'OM_Process' to hold this information. However, OM_Process is an abstract Class; we need a concrete implementation.

The Procedure shall be used to define a set of actions that are consistent across a set of related Observations; one must reference a different OM_Process instance if the actions (including any configuration settings) would lead to a different interpretation of the Observation result. Thus communities define a broad range of entities for OM_Process - from a specific instrument with a specific calibration to a textual description of a routine process.

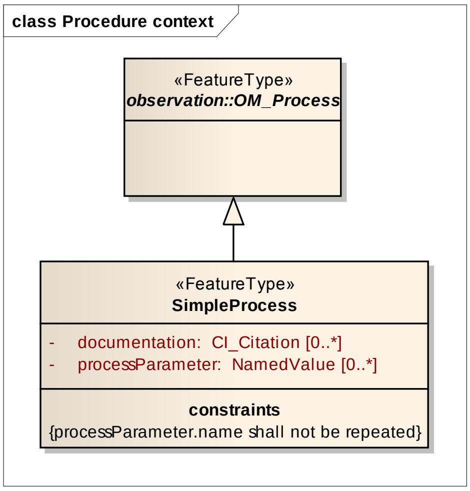
Examples of information detailing the procedure include:

- Table B Class 02 'Instrumentation' code [0 02 001] 'Type of station (AUTO)'
- Table B Class 07 'Location (vertical)' code [0 07 030] 'Height of station ground above mean sea level'[18]
- Table B Class 07 'Location (vertical)' code [0 07 031] 'Height of barometer above mean sea level'
- Table B Class 07 'Location (vertical)' code [0 07 032] 'Height of sensor above local ground'[19]

A simple Class has been developed to specialise OM_Process that can be used to reference on-line documentation and process-specific parameters (e.g. sensorHeight).

[18] At WMO TT-AvXML [M1] we agreed that Aviation would use 2.5Dimensional models (e.g. the vertical dimension would be treated as property of the model rather than specifying coordinates in {x, y, x})

[19] this code appears twice in the METAR/SPECI template - once for wind-related observations (where the default value is set to 10m), and later for temperature-related observations (where default value is set to 2m)



CI_Citation (from ISO 19115 - Metadata) is a reasonably complicated Class - however, the majority of the attributes are optional. An instance of SimpleProcess might be as follows:

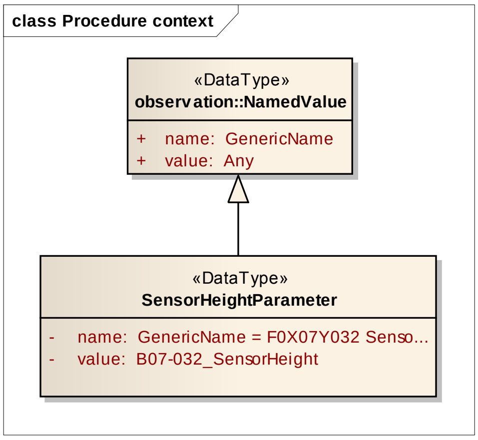
```
<procedure>
  <SimpleProcess>
    <documentation>
      <CI_Citation>
        <title>
          <CharacterString>WMO No. 8 Part 1-2 Measurement of temperature</CharacterString>
        </title>
        <onlineResource>
          <CI_OnlineResource>
            <linkage>http://www.wmo.int/pages/prog/www/IMOP/publications/.../ Chapter%202.pdf</linkage>
          </CI_OnlineResource>
        </onlineResource>
      </CI_Citation>
    </documentation>
    <processParameter>
      <name>F0X02X001 Station Type</name>
      <value>
        <B02-001_StationType
          codeList="http://{hostname}/{path}/BUFRCREX/Code+Flag/02.001"
          codeListValue="http://{hostname}/{path}/BUFRCREX/Code+Flag/02.001/1">
          Manned
        </B02-001_StationType>
      </value>
    </processParameter>
    <processParameter>
      <name>F0X07X030 Station Height</name>
      <value>
        <!-- [0 07 030] 'Height of station ground above mean sea level' -->
        <B07-030_StationHeight nilReason="missing"/>
      </value>
    </processParameter>
    <processParameter>
      <name>F0X07X032 Sensor Height</name>
      <value>
        <!-- [0 07 032] 'Height of sensor above local ground (or deck of marine platform)' -->
        <B07-032_SensorHeight nilReason="missing"/>
      </value>
    </processParameter>
  </SimpleProcess>
</procedure>
```

It is essential that the Process 'documentation' describes each of the named 'processParameters' - including provision of units of measure as necessary (e.g. 'sensorHeight' [0 07 032]'Height of sensor above local ground (or deck of marine platform)' is designated with units = 'm' ... the BUFR Table B Class 07 provides the definition of this parameter as indicated above).

The process documentation shall need publishing with definitive HTTP URI.

Thus the codes that define configuration for the Procedure ([0 02 001] 'Type of station (AUTO)', [0 07 030] 'Height of station ground above mean sea level', [0 07 031] 'Height of barometer above mean sea level' and [0 07 032] 'Height of sensor above local ground') shall need to be published with definitive (HTTP) URIs - within a Registry that will provide useful information when those URIs are resolved. The process documentation shall indicate the full set of anticipated parameters (not all parameters may be specified in an instance document) and their default values[20].

It may be prudent to provide specialisations of NamedValue for each of these terms to simplify validation; e.g.



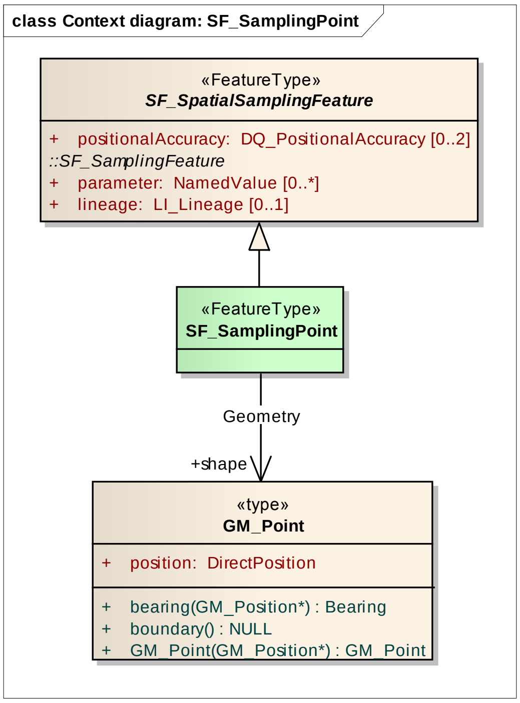
B07-031_SensorHeight specialises the 'Distance' Measure - and thus will carry the appropriate unit of measure information.

Note: where Measures such as [0 12 011] 'Maximum temperature, at height and over period specified' are used, it is essential to pre-set the qualifiers before the data values are specified. One place where this should occur is within the Process. In this case, the sensorHeight [F0X07031 SensorHeight] and period [F0X04Y074 Period] should be included as parameters in the SimpleProcess instance.

Modelling the Observation featureOfInterest

CSML3:Point Observation constrains the type of Class referred to via OM_Observation/featureOfInterest to be SF_SamplingPoint:

[20] default values cannot be provided in the register entry for, say, [0 07 032] as the value depends on its context of use - e.g. which procedure it is being used with

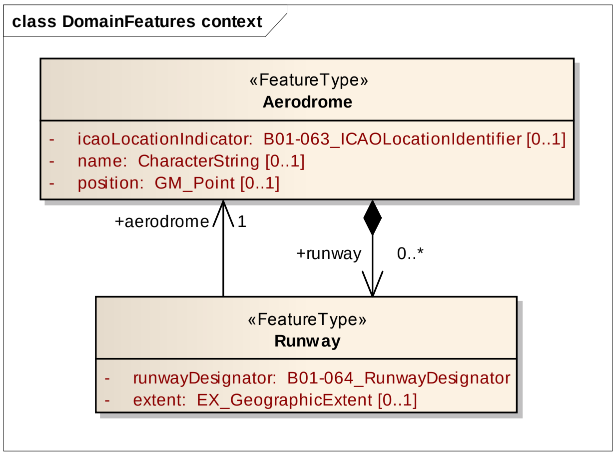


SF_SamplingPoint has a number of attributes - many inherited from super-Classes:

- shape:GM_Point - used to define the spatial manifold of the observation result data (in this case, a simple geometric point)
- positionalAccuracy:DQ_PositionalAccuracy[0..2] - which may be used to indicate the accuracy of the coordinates defining the 'shape'
- lineage:LI_Lineage[0..1][21]
- parameter:NamedValue[0..*]
- sampledFeature:GFI_Feature - which can be used to reference a 'domain feature' such as Aerodrome or Runway. These domain objects should be available to reference (e.g. pre-published as part of the ICAO SWIM & referenced via URI). Within BUFR encoding this property may be constrained to the appropriate Table B Class 01 'Identification' type (e.g. [0 01 063] 'ICAO location indicator' or [0 01 064] 'Runway designator').
- ...

The attribute ‘sampledFeature’ is used to reference domain features – such as a specific Aerodrome or Runway. We anticipate that these will be provided by ICAO. In the interim, simple Classes are defined for Aerodrome and Runway based on examples from WXXM1.1.

[21] At some point in the future, the ‘lineage’ element may be used to describe the history of a given Observation Station



Grouping Measures into Observations

The WXXM model defines the METAR as a 'Report' which contains multiple Observations. This approach is also deployed within the WMO Logical Data Model.

However – just how many OM_Observation instances will the METAR/SPECI Report contain?

Our previous analysis of the METAR/SPECI BUFR Template was able to define a series of distinct Classes based on the presence of Generalised Coordinates, Significance Qualifiers and replication elements.

Furthermore, we note that the main part of the METAR/SPECI Report and the Trend Forecast specify the Aerodrome as the featureOfInterest, whilst the runway-related observations clearly specify a Runway as featureOfInterest. The change of featureOfInterest is a clear discriminator enabling discretisation of Observations.

Finally, looking at the resulting model, it seems appropriate to cluster adjacent Properties from the same BUFR Table B Class into a single OM_Observation; each of these Properties are likely to share similar observing procedures (*or at least it makes sense to group all similar observation steps into a single composite process!*)

Thus Observations may be grouped as follows:

Main Aerodrome observation elements

- 0 01 063 ICAOLocationIndicator
    - StationType/Date/Time/Position
        - 0 07 032 SensorHeight

        [Class 11 'Wind and turbulence']

            - 0 11 001 WindDirection
            - 0 11 016 VariableWindDirectionCCW
            - 0 11 017 VariableWindDirectionCW
            - {WindSpeedGroup}

            [Class 11 'Wind and turbulence']

                - 0 08 054 WindSpeedOrGustQualifier
                - 0 11 083 WindSpeed_kmh-1
                - 0 11 084 WindSpeed_knots
                - 0 11 002 WindSpeed_ms-1

            - {WindGustGroup}

[Class 11 'Wind and turbulence']

        - 0 08 054 WindSpeedOrGustQualifier
        - 0 11 085 MaxWindSpeed_kmh-1
        - 0 11 086 MaxWindSpeed_knots
        - 0 11 041 MaxWindSpeed_ms-1
    - 0 07 032 SensorHeight

    [Class 12 'Temperature']

        - 0 12 023 Temperature
        - 0 12 024 Dewpoint
    - 0 07 031 BarometerHeight

    [Class 10 'Non-coordinate location - vertical']

        - 0 10 052 QNH

[Class 20 'Observed phenomena']

- 0 20 009 WeatherIndicator
- 0 20 060 PrevailngVisibility
- {x 0..2} DirectedMinVisbility
    - 0 05 021 Bearing

    [Class 20 'Observed phenomena']

        - 0 20 059 MinVisibility
- 0 20 019 PresentWeather
- {x 0..3} Clouds
    - 0 08 002 VerticalSignificance

    [Class 20 'Observed phenomena]

        - 0 20 011 CloudAmount
        - 0 20 012 CloudType
        - 0 20 013 CloudVaseHeight_m
        - 0 20 092 CloudBaseHeight_ft
- 0 20 002 VerticalVisibility_m
- 0 20 091 VerticalVisibility_ft
- 0 20 020 RecentWeather

[Class 11 'Wind and turbulence']

Comment [J43]: Should this be merged with the earlier Class 11 elements?

- 0 11 070 WindShearAffectedRunway
- {x 0..1} SeaConditions

    [Class 22 'Oceanographic elements']

    - 0 22 043 WaterTemperature
    - 0 22 021 WaveHeight

[Class 20 'Observed phenomena']

- 0 20 085 RunwayCondition

Runway observation elements

- {x 0..4} RunwayVisualRangeGroup
    - 0 01 064 RunwayDesignator
        - {x2}
            - 0 08 014 RVR Qualifier

[Class 20 'Observed phenomena']

- 0 20 061 RVR

[Class 20 'Observed phenomena']

- 0 20 018 RVRTendency

- {x 0..4} RunwayConditionSummary
  - 0 01 064 RunwayDesignator

    [Class 20 'Observed phenomena']
    - 0 20 085 RunwayCondition
- {x 0..4} RunwayConditionDetails
  - 0 01 064 RunwayDesignator

    [Class 20 'Observed phenomena']
    - 0 20 086 RunwayDeposits
    - 0 20 087 RunwayContamination
    - 0 20 088 RunwayDepositDepth
    - 0 20 089 RunwayFrictionCoefficient

Trend forecast elements

- {x0..3} TrendForecast
  - 0 08 016 TrendForecastChangeQualifier
    - Time
      - 0 07 032 SensorHeight

      [Class 11 'Wind and turbulence']
      - 0 11 001 WindDirection
      - {WindSpeedGroup}

        [Class 11 'Wind and turbulence']
        - 0 08 054 WindSpeedOrGustQualifier
        - 0 11 083 WindSpeed_kmh-1
        - 0 11 084 WindSpeed_knots
        - 0 11 002 WindSpeed_ms-1
      - {WindGustGroup}

        [Class 11 'Wind and turbulence']
        - 0 08 054 WindSpeedOrGustQualifier
        - 0 11 085 MaxWindSpeed_kmh-1
        - 0 11 086 MaxWindSpeed_knots
        - 0 11 041 MaxWindSpeed_ms-1

      [Class 20 'Observed phenomena']
      - 0 20 009 WeatherIndicator
      - 0 20 060 PrevailingVisibility
      - 0 20 019 PresentWeather
      - {x 0..3} Clouds
        - 0 08 002 VerticalSignificance

        [Class 20 'Observed phenomena]
          - 0 20 011 CloudAmount
          - 0 20 012 CloudType

    - 0 20 013 CloudVaseHeight_m
    - 0 20 092 CloudBaseHeight_ft
- 0 20 002 VerticalVisibility_m
- 0 20 091 VerticalVisibility_ft

Note: one can represent the observation information from each Runway as a separate Observation - with the Runway domain feature referenced as the sampledFeature. An alternative would be to push the runwayDesignator into the CoverageRecord and retain the Aerodrome domain feature as the sampledFeature. Currently we describe the Aerodrome as a single geometric point (expressed in coarse resolution latitude and longitude) - thus we use a PointObservation with an SF_SamplingPoint. However, an Aerodrome clearly has physical extent, and each Runway will have a different position (which notionally may be described as a geometric point - or at least the notional position where the observations are taken). If we collect the Measures from ALL Runways in a particular Aerodrome together in a single coverage, we should use MultiPointObservation with SF_SamplingSurface describing the extent of the Aerodrome; the notional geometry of each Runway shall be used to express the Coverage domain. Examples of this method have not been implemented. Also note, that this would require Class D07-XXX_RunwayObservedPhenomenaRecord to be modified to include the runwayDesignator element (B01-064_RunwayDesignator).

AerodromeObservations

Thus the 'main' set of Measures for the METAR would be grouped into 7 Observations:

[Class 11 'Wind and turbulence'] - D07-403_WindRecord
    0 11 001 WindDirection
    0 11 016 VariableWindDirectionCCW
    0 11 017 VariableWindDirectionCW
    {WindSpeedGroup} 0 08 054 WindSpeedOrGustQualifier
    [Class 11 'Wind and turbulence']
        0 11 083 WindSpeed_kmh-1
        0 11 084 WindSpeed_knots
        0 11 002 WindSpeed_ms-1
    {WindGustGroup} 0 08 054 WindSpeedOrGustQualifier
    [Class 11 'Wind and turbulence']
        0 11 085 MaxWindSpeed_kmh-1
        0 11 086 MaxWindSpeed_knots
        0 11 041 MaxWindSpeed_ms-1

[Class 12 'Temperature'] - D07-404_TemperatureRecord
    0 12 023 Temperature
    0 12 024 Dewpoint

[Class 10 'Non-coordinate location - vertical'] - D07-405_QNHRecord
    0 10 052 QNH

[Class 20 'Observed phenomena'] - D07-406_ObservedPhenomenaRecord
    0 20 009 WeatherIndicator
    0 20 060 PrevailngVisibility
    {x 0..2} DirectedMinVisbility
        0 05 021 Bearing
        0 20 059 MinVisibility

0 20 019 PresentWeather
{x 0..3} Clouds
0 08 002 VerticalSignificance
0 20 011 CloudAmount
0 20 012 CloudType
0 20 013 CloudBaseHeight_m
0 20 092 CloudBaseHeight_ft
0 20 002 VerticalVisibility_m
0 20 091 VerticalVisibility_ft
0 20 020 RecentWeather

[Class 11 'Wind and turbulence'] - D07-407_RunwayWindShearRecord
0 11 070 WindShearAffectedRunway

[Class 22 'Oceanographic elements'] - D07-408_SeaConditionsRecord
0 22 043 WaterTemperature
0 22 021 WaveHeight

[Class 20 'Observed phenomena'] - D07-409_RunwayConditionRecord
0 20 085 RunwayCondition

RunwayObservations

The set of Measures for **each** Runway would be grouped into a single Observation (noting that there may zero to four Runways described):

[Class 20 'Observed phenomena'] - D07-411_RunwayObservedPhenomenaRecord
{0..1}
{x2}
0 08 014 RVR Qualifier
0 20 061 RVR
0 20 018 RVRTendency
{0..1}
0 20 085 RunwayCondition
{0..1}
0 20 086 RunwayDeposits
0 20 087 RunwayContamination
0 20 088 RunwayDepositDepth
0 20 089 RunwayFrictionCoefficient

TrendForecast

The set of Measures for each trend forecast would be grouped into 2 Observations (noting that there may be zero to three trend forecasts described):

[Class 11 'Wind and turbulence'] - D07-415_ForecastWindRecord
0 11 001 WindDirection
{WindSpeedGroup} 0 08 054 WindSpeedOrGustQualifier
0 11 083 WindSpeed_kmh-1
0 11 084 WindSpeed_knots

0 11 002 WindSpeed_ms-1
{WindGustGroup} 0 08 054 WindSpeedOrGustQualifier
0 11 085 MaxWindSpeed_kmh-1
0 11 086 MaxWindSpeed_knots
0 11 041 MaxWindSpeed_ms-1

[Class 20 'Observed phenomena'] - D07-416_ForecastObservedPhenomenaRecord
0 20 009 WeatherIndicator
0 20 060 PrevailingVisibility
0 20 019 PresentWeather
{x 0..3} Clouds
0 08 002 VerticalSignificance
0 20 011 CloudAmount
0 20 012 CloudType
0 20 013 CloudBaseHeight_m
0 20 092 CloudBaseHeight_ft
0 20 002 VerticalVisibility_m
0 20 091 VerticalVisibility_ft

A soft-typed approach: constraining Observation Classes

Each group of measures shall be expressed in a single Observation.

We seek to re-use the PointObservation Class throughout the METAR/SPECI Product. It must be constrained to ensure that the correct properties are included.

Each Observation shall be constrained as follows:

- Ensure the 'sampledFeature' is of the correct type:

  {observation}.featureOfInterest.sampledFeature.oclIsKindOf.(*{Aerodrome|Runway}*)… e.g. qnh.featureOfInterest.sampledFeature.oclIsKindOf.(Aerodrome)

- Ensure the correct Process Parameters are included:

  {observation}.procedure.processParameter[n].oclIsKindOf(*{Process Parameter Type}*) … e.g. qnh.procedure.processParameter[0].oclIsKindOf(BarometerHeightParameter)

- Ensure the correct Record type is used[22]:

  {observation}.result.element[*].value.oclIsKindOf(*{Record Type}*) … e.g. qnh.result.element[*].value.oclIsKindOf(D07-405_QNHRecord)

Each of these Observations shall be of type PointObservation; this implies the application of yet more constraints.

(product status may be implemented as an XML attribute)

>>>

Defining the METAR/SPECI Report

[22] pre-existing constraints from CV_Coverage shall ensure that the Record matches the description provided by 'rangeType'

A number of elements appear to apply to the METAR report itself - or sections thereof:

- Table B Class 08 code [0 08 079] 'Aviation product status (routine, special, corrected, not available) METAR SPECI COR'
- Table B Class 08 code [0 08 016] 'Change qualifier for trend type forecast TTTTT NOSIG' … *each of the zero to three trend forecasts are qualified in this way*

These elements shall be included in the 'report' structure within which the Observations are packaged.

The Observations shall be arranged in three collections:

- aerodromeObservations [1] … containing up to 7 Observations
- runwayObservation [0..4] … each containing a single Observation
- aerodromeTrendForecast [0..3] … each containing 2 Observations

Where multiple Observations are clustered, a Collection Class shall be used:

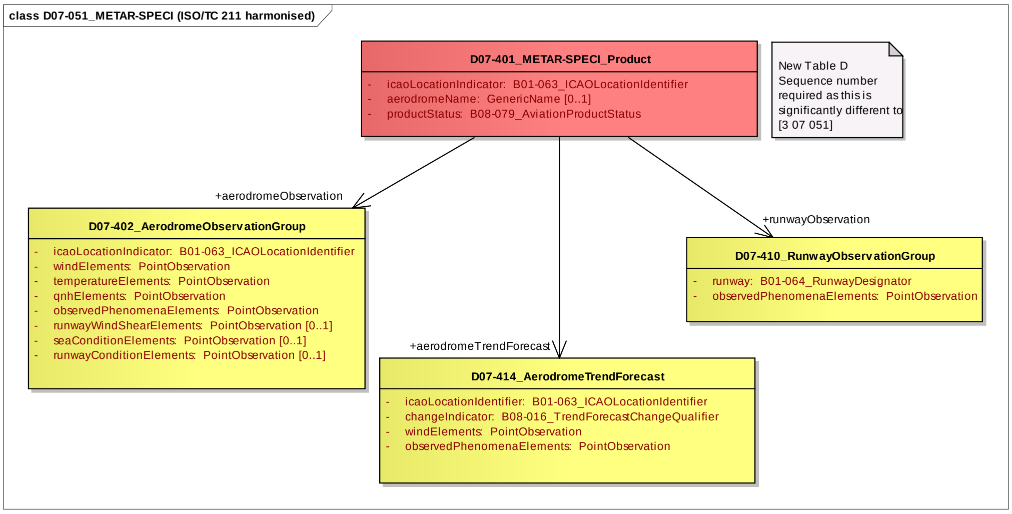
- aerodromeObservations:D07-402_AerodromeObservationGroup [1]
    - icaoLocationIdentifier:B01-063_ICAOLocationIdentifier
    - wind:PointObservation
    - temperature:PointObservation
    - qnh:PointObservation
    - observedPhenomena:PointObservation
    - runwayWindShear:PointObservation [0..1]
    - seaConditions:PointObservation [0..1]
    - runwayCondition:PointObservation [0..1]
- aerodromeTrendForecast:D07-414_AerodromeTrendForecast [0..3]
    - icaoLocationIdentifier:B01-063_ICAOLocationIdentifier
    - changeIndicator:B08-016_TrendForecastChangeQualifier
    - wind:PointObservation
    - observedPhenomena:PointObservation

This pattern is repeated for symmetry with runwayObservation -

- runwayObservation:D07-410_RunwayObservation [0..4]
    - runway:B01-064_RunwayDesignator
    - observedPhenomena:PointObservation

The main METAR/SPECI Report (D07-XXX_METAR-SPECI_Report) shall include:

- icaoLocationIndicator:B01-063_ICAOLocationIndicator
- aerodromeName:GenericName [0..1]
- productStatus:B08-079_AviationProductStatus
- aerodromeObservation:D07-402_AerodromeObservation
- runwayObservation:D07-410_RunwayObservation [0..4]
- aerodromeTrendForecast:D07-414_AerodromeTrendForecast [0..3]


class D07-051_METAR-SPECI (ISO/TC 211 harmonised)
D07-401_METAR-SPECI_Product
- icaoLocationIndicator: B01-063_ICAOLocationIdentifier
- aerodromeName: GenericName [0..1]
- productStatus: B08-079_AviationProductStatus
New Table D Sequence number required as this is significantly different to [3 07 051]
+aerodromeObservation
D07-402_AerodromeObservationGroup
- icaoLocationIndicator: B01-063_ICAOLocationIdentifier
- windElements: PointObservation
- temperatureElements: PointObservation
- qnhElements: PointObservation
- observedPhenomenaElements: PointObservation
- runwayWindShearElements: PointObservation [0..1]
- seaConditionElements: PointObservation [0..1]
- runwayConditionElements: PointObservation [0..1]
+runwayObservation
D07-410_RunwayObservationGroup
- runway: B01-064_RunwayDesignator
- observedPhenomenaElements: PointObservation
+aerodromeTrendForecast
D07-414_AerodromeTrendForecast
- icaoLocationIdentifier: B01-063_ICAOLocationIdentifier
- changeIndicator: B08-016_TrendForecastChangeQualifier
- windElements: PointObservation
- observedPhenomenaElements: PointObservation

Annex 1: implications for BUFR Edition 5

This is section contains un-validated thoughts on implications for BUFR Edition 5.

- Lexical processing model.
- (POLYMORPHISM) Establish a mechanism that permits substitution of a specific Sequence for a more general Sequence - thus matching the behaviour of substituting a sub-Class for a more general super-Class within the UML model (e.g. using D07-405_QNHRecord in place of Record)[23]
- Establish a mechanism for implementing the UML constraints - although at present I don't know whether this would form constraints on a BUFR Sequence or require the 'constrained' Sequence to be documented explicitly.
- Determine any potential implications of the UML Packaging structure with respect to namespaces of elements defined therein (I think each Package might map to a different Table D Class - implying we may need new Table D Classes)

Additionally, new Table D Sequences will be required to:

- encode the Classes defined herein (e.g. ObservableProperty, SimpleProcess etc.)
- Explicit closure of Sequences by setting Generalised Coordinates & Significance Qualifiers to 'missing'

[23] An alternative option is to explicitly expand all the Sequences into a complete template; this _MIGHT_ be achievable by analysing the constraints … ?

Annex 2: outstanding actions from the UML model

- Check correct use of Class «stereotypes»
- Import 'Extended UML metamodel for XML Schema' and assign tagged values as appropriate for conversion to XML Schema via Fullmoon
- Assess which UML attributes should be expressed as XML attributes
- Assignment of tagged values
- Check correct use of units of measure 'default' values in Table B Measure Classes … (*also check that XML examples use correct UCUM terms*)
- Check correct syntax of constraints (OCL)
- Add metadata from BUFR Section 1 into the METAR Report; e.g.
    - BUFR Code table A 'Data category' (e.g. [0] 'Surface data - land')
    - Common Code table C.13 'Data sub-category' (e.g. [0.10] 'Routine aeronautical observations (METAR)')
    - Message publication date-time
    - Originating centre / sub-centre … Common Code table C.11
- __PACKAGING__ structures

Once we're happy with this approach, we need to apply this to TAFs and SIGMETs … and to SYNOPs to verify the approach for non-aviation data.

Further analysis of more complex BUFR Templates will be required – especially those that routinely qualify properties. For example, the TEMP record is a good example to test our assumptions. Here, one expects the following Sequence:

- F0X08Y001: vertical sounding significance {surface|standard level|tropopause level|maximum wind level|… etc ...}
- F0X07Y___: height, geopotential height, pressure etc. (to indicate *which* standard level)
- F0X12Y001: temperature
- ...

The vertical sounding significance actually changes the meaning of the temperature: tropopause temperature _MUST NOT_ be mixed with surface temperature: don't mix tropopause temperatures and surface temperatures in the same Observation instance.

Annex 1: example METAR report:

```
\\Section 4
232        \ Length of section
0          \ Reserved (1 octet)
LKKV       \ 001063 ICAO LOCATION INDICATOR [CCITTIA5]
0          \ 008079 Aviation PRODUCT [Code]
           \        Normal issue
1          \ 002001 TYPE OF STATION [Code]
           \        Manned station
2007       \ 004001 YEAR [year]
7          \ 004002 MONTH [month]
25         \ 004003 DAY [day]
12         \ 004004 HOUR [hour]
0          \ 004005 MINUTE [minute]
/          \ 005002 LATITUDE (COARSE ACCURACY) [degree]
/          \ 006002 LONGITUDE (COARSE ACCURACY) [degree]
/          \ 007030 HEIGHT OF STATION GROUND ABOVE MEAN SEA LEVEL [m]
/          \ 007031 HEIGHT OF BAROMETER ABOVE MEAN SEA LEVEL [m]
/          \ 007032 HEIGHT OF SENSOR ABOVE LOCAL GROUND [m]
210        \ 011001 WIND DIRECTION [DEGREE TRUE]
/          \ 011016 EXTREME COUNTERCLOCKWISE WIND DIRECTION OF A VARIABLE WIND
[DEGREE TRUw
/          \ 011017 EXTREME CLOCKWISE WIND DIRECTION OF A VARIABLE WIND [DEGREE
TRUE]
0          \ 008054 Qualifier for wind speed or wind gusts [Code]
/          \ 011083 WIND SPEED [km/h]
5          \ 011084 WIND SPEED [knot]
2.6        \ 011002 WIND SPEED [m/sec]
0          \ 008054 Qualifier for wind speed or wind gusts [Code]
/          \ 011085 MAXIMUM WIND GUST SPEED [km/h]
/          \ 011086 MAXIMUM WIND GUST SPEED [knot]
/          \ 011041 maximum WIND SPEED (GUSTS) [m/sec]
/          \ 008054 Qualifier for wind speed or wind gusts [Code]
/          \ 007032 HEIGHT OF SENSOR ABOVE LOCAL GROUND [m]
27         \ 012023 Temperature [C]
10         \ 012024 DEW POINT temperature [C]
/          \ 007032 HEIGHT OF SENSOR ABOVE LOCAL GROUND [m]
10100E1    \ 010052 ALTIMETER SETTING (QNH) [Pa]
2          \ 020009 GENERAL WEATHER INDICATOR [Code]
           \        CAVOK
/          \ 020060 Prevailing HORIZONTAL VISIBILITY [m]
0          \  031001 Delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
/          \ 020002 VERTICAL VISIBILITY [m]
/          \ 020091 VERTICAL VISIBILITY [ft]
0          \  031001 Delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
0          \  031000 Short delayed descriptor replication factor []
0          \  031000 Short delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
0          \  031001 Delayed descriptor replication factor []
```