WORLD METEOROLOGICAL ORGANIZATION

COMMISSION FOR BASIC SYSTEMS

OPAG ON INFORMATION SYSTEMS & SERVICES

EXPERT TEAM ON ASSESSMENT OF DATA REPRESENTATION SYSTEMS

FIRST MEETING

WASHINGTON, USA, 23 TO 25 APRIL 2008

ET-ADRS-1/ Doc. 4.1(3)
(07.04.2008)

ITEM 4

ENGLISH only

World Meteorological Organisation Spatio-temporal Data Standards Similarity to ISO 19100 Geographic Information Standards

(Submitted by Mr Gil Ross (UK))

Summary and purpose of document

This paper describes the model of WMO BUFR, and explains commonalities and differences with the ISO 19100 standards for Geographic Information Standards.

ACTION PROPOSED

The ET-ADRS is invited to consider the information contained in the document concerning the model of WMO BUFR, and commonalities and differences with the ISO 19100 standards for Geographic Information Standards, in particular with a view to developing a WMO conceptual model of data representation, as a fundamental element of a CBS policy on data representation systems.

World Meteorological Organisation Spatio-Temporal Data Standards

Similarity to ISO 19100 Geographic Information Standards

Gil Ross   Met Office

1. Introduction:

WMO, as the specialist UN Agency for Meteorology, have been sponsoring data standards since the UN passed a Directive in 1963 which set up the World Weather Watch.

Initially the telecommunications methods were teletype and radio-transmission via Morse code. So all the initial data codes were human readable (in that they were coded, transmitted and decoded by humans) but they were highly formalised and based on hand operated Morse standards, mainly organised in 5 groups.

These Traditional Alphanumeric Codes (TACs) are still in full use. Hundreds of thousands of messages with these codes are received every day from every corner of the globe. There are around 50 such code forms defined in the WMO Volume 306, Manual of Codes Part A, and these are still assiduously maintained by the WMO Expert Team on Data Representation and Codes.

The replacements for the TAC forms are also in wide use. These are commonly called “Table Driven Code Forms”, because both the elements and the non-numeric values which the codes contain are predefined in many tables (at least 450 tables for those maintained by ET DR&C). There is a program to migrate individual TAC forms to table driven codes, but there are still valid reasons for using the TAC forms – they can be written and read by humans. Very many people, particularly in aviation, still have the skills to read these codes. There are possibly 2-3 million pilots, no matter what language they use, who read the METAR and TAF codes by which weather conditions and forecasts for airports are sent to en-route aircraft each and every hour.

In 1989, FM92 GRIB (GRIdded Binary) was accepted as a new format for gridded 2 dimensional scalar data , and is now extended to multi-dimensional tensor data. This has almost entirely replaced the original TAC gridded codes because of enormously superior data compression capabilities.

At the same time the binary code FM94 BUFR (Binary Universal Form for Representation) was introduced for discrete (and mostly observational) data, and the non-binary version FM95 CREX (Character form for the Representation and EXchange of data) was approved in 1994, but only went operational in 2000.

This paper describes the model of WMO BUFR, and explains commonalities and differences with the ISO 19100 standards for Geographic Information Standards. Since CREX is essentially a character representational form of BUFR, and so it is implicitly included. GRIB however is not considered here because, while many of the concepts explored apply to GRIB too, GRIB is also a simpler model than BUFR. It is only where it develops concepts of multidimensional grids, that GRIB extends the modelling framework.

2. BUFR – as seen by the designers.

Although the TAC codes are tightly defined and over the years have been rigorously maintained, they require the use of specific decoders in any automated system. The decoders seldom were designed for separation of the business rules and the decoding application. So the information in the TAC forms is complicated to modify, if only because there are many decoders which need to be modified too.

Although the BUFR design wasn't quite described in the following terms, the expert team had a number of design requirements.

2.1 BUFR Design Requirements

BUFR was to be a general purpose system for exchanging data. To do that, the information content and the coding forms including the (de)coding software were to be separated.

- This meant that BUFR was designed to be much more flexible and expandable than TAC forms.

- BUFR information was to be fully self-describing. By this, it was meant that both the form and content of the information in a BUFR message was to be contained in the message itself. (BUFR was being developed at the same era as SGML, but it is not clear whether there was any contact between the projects)

- BUFR was to be extremely well compressed. This is the antithesis to the SGML offspring XML, which has a design constraint that:
  "Terseness in XML Markup is of minimal importance."

- An important design goal was to make the "obvious" explicit. This meant exclude as much "self evident" information as possible, instead requiring that "meaning" be specified by definite rules.

The way in which these contradictory requirements were achieved, and the enormous scale, completeness and authority of the BUFR modelling, is impressive indeed.

The result however has never been clearly described and much of BUFR documentation is recondite. Although the BUFR documentation is clearly published and the BUFR tables equally so (see the references in chapter 6), the documentation is unyielding. While there is software freely available, it is aimed at the specialist. BUFR has a poor reputation outside of operational meteorology.

In part the documentation is complicated because the BUFR model is complex, but it also reflects the practices of the era in which BUFR was designed. In 1988, the emphasis was on how to code and decode BUFR – NOT on the modelling process, and it is on BUFR decoding that most of the documentation focuses.

2.2 How Compression and Self-Description work together.

Rather than describing in detail, the coding/decoding process, here we shall discuss the principles behind the algorithms.

- The two, seemingly contradictory, concepts of compression and the self-description in BUFR are gained by making everything, tags and values (except number values), a reference. The references are indexes to elements[1] (rows) of published tables.

- The tags – element descriptors – and the values are grouped apart. When expanded, all the tags are in sequence with all the values, but the tags are grouped separately from the values.

- Number values are compressed by closely defining the precision, scaling by an exponent and removing any reference value for the mantissa for each element in the tables. This makes every number an unsigned integer stored in a predefined bit length binary.

- As well as each number-value element having predefined compression, there are dynamic redefinition operators, by which temporary precisions can be defined.

- Some common elements which could be subject to varying precisions are duplicated as a different element, but are really only the same parameter with a different precision.

- Further compression is gained by explicit grouping of terms in predefined collections of elements in a template reference. This is made modular by allowing collections to refer to other collections of templates.

- There are dynamic replication mechanisms, which in conjunction with the templates or collections can allow a variable replication of element sequences. Similarly there are dynamic repetition mechanisms to allow repeated values to be compressed in effectively a run-length encoding.

[1] Many of the terms used here have specific meaning, and are likely to vary across different modelling themes. Database modelling, UML modelling, XML modelling, different programming paradigms all use terms such as element, row, attribute, descriptors, tables even, arrays, lists etc. with different meanings. Contrariwise, identical concepts are often assigned different names in different canons.

- Yet further compression and enormous flexibility can be achieved by developing a dynamic grouping mechanism which BUFR calls a generalised “coordinate” mechanism. In application this is a single “grouping” mechanism to assign properties to a group of features. In ISO terms, this generalised coordinate can be considered at the same time as a feature collection mechanism, a coverage grid mechanism, and a feature attribute mechanism. These generalised coordinates can be horizontal position, or they can be identifiers, declarations of common instrumentation, temporal or (generalised) vertical coordinates or they can be “significance” qualifiers.

- There are also packing mechanisms in which array structures can have dynamic offsets removed to reduce the bit length of individual members.

There are other mechanisms which have been added over the years for different tasks, which have complicated further the BUFR mechanisms. As with many standards, the added functionality makes the maintenance much more difficult, and leads to confusion, not clarification.

However, it seems than many of the added mechanisms are niche requirements. Although it isn’t declared so in the documentation, from sampling the actual practical implementation in BUFR messages, few of the added mechanisms were intended to work together. Where complications are used, they seem to be used in isolation.

This means that some of the complications added in later years can be ignored here as they are coding/decoding complexities, and are not important from a data modelling perspective.

3. The WMO BUFR standard.

BUFR was designed for any data type which could be fully described in a formal table structure. It is primarily a code table or fixed format numeric type, where the values are individual. Although it can also code text strings and arrays, it is not efficient for these. For multi-dimensional array data, the WMO GRIB format is considerably better.

However, the mechanism for table creation allows the message to specify what tables are used, who published them and in what version. The BUFR model also allows the BUFR tables to be self described, and BUFR messages can be used to exchange new versions of the BUFR codes, themselves coded in BUFR.

As well as the ability to specify the whole table structure, there are also mechanisms to allow local definitions within the specification, or to modify many of the global specifications temporarily.

The main WMO table sets are published on the web (see references in chapter 6), as are many locally defined tables and extensions, but given the period when BUFR was designed, the publications are static documents, intended to be hand extracted and implemented into bespoke software by specialists.

3.1 BUFR Tables, references and indexes

BUFR tables are the response to many requirements. The tables are organised for separate purposes initially, but some of the model functions and some coding functions are spread between the table types.

3.1.1 BUFR Table types

Firstly there are Common Coding Tables, where definitions used across different code forms (specifically BUFR and GRIB) are defined. These tables are external to both the model and coding functions, and for example, list all the data creation authorities across WMO.

Table A has the function of defining the sub-catalogue type as part of the full BUFR feature catalogue. The 20 sub-catalogues represent different high-level use cases, but this level of categorisation is only partially followed in the collections of features and classes of simple features, because of re-use of the lower level classes.

Table B contains the list of simple features. These are grouped into 30 classes at present, and there are around 1200 simple features in these classes. At a higher level, the first 9 classes hold the BUFR coordinates, which is a grouping property and by which the BUFR process akin to ISO coverage grids are defined. Table B is used at both the modelling and coding levels.

Table C contains coding operators. Table C also allows some dynamic feature properties, such as annotation and associated features, but mainly it modifies the attributes of existing, more static Table B features.

Table D defines collections of Table B features, and being modular, collections of other Table D collections. Table D is also the templating table, by which exchange formats are predefined.

3.1.2 BUFR Indexes

Table B, C and D and a single row "table", the replication operator are coded up with an indexing algorithm. All tags in a BUFR message or instance (rows in the tables and code tables, described below) are referenced by this indexing algorithm. The index has three parts (replace Z with B, C or D):

- **F:** Table_**Z**_indicator: is descriptor (row, tag or element) type. This takes the value 0 for Table B references, 1 for the replicator operator, 2 for the coding operators in Table C, and 3 for the "sequence" descriptors, the collections or templates of Table D.
  F occupies 2 bits and can have 4 values
- **X:** Table_**Z**_elementClass: is the sub-class of Table B, C or D tables, and a "number of terms" for the replication operator.
  X occupies 6 bits and can have 64 values.
- **Y:** Table_**Z**_codeValue: references the individual descriptor, feature or feature collection in Tables B and D, and the operator in Table C. For the replication operator it is the number of replications to be performed.
  Y occupies 8 bits and can have 256 values.

In Table B, features are assigned base units and precisions. The units are all in SI units, e.g. all in Kelvin for temperature. For discrete features or coverages, the unit may refer to an enumeration – called a code or flag table. There are 370 code and flag tables, and all are referenced by the same index that their table B feature has.

A code table in WMO terms is an exclusive enumeration which may or may not be complete (e.g. not complete – means that there may be reserved code values, exclusive here means that only one value is valid at the same time). A flag table is usually complete, but is not exclusive, i.e. multiple flag values are allowed. Flag values are coded as powers of 2, so that a bit pattern can hold simultaneous flags.

3.1.3 BUFR references within a BUFR instance

A BUFR message or instance references a code and version number for which BUFR tables are being used – the WMO current tables or local tables. From then on all the collections, elements or operators are referenced by index F-X-Y coded in 16 bits.

To decode the BUFR instance, the references are followed through, e.g. sequences or collections are expanded, operations are performed, fixed length non-negative integers are transformed, and code values are imported from the code tables.

This gives a list of simple features identifiers, and an equivalent list of values or code values to which they map. Coordinate groupings and collection groupings can give the tree hierarchy of features, and a further mapping can expand feature names, code value meanings, parameter units etc..

This is entirely equivalent to a full specification in a markup language, where the specific document schema travels with the document, like a DTD contrasted with an XSD schema.

3.2 BUFR UML

The WMO BUFR standard is described here with the aid of UML diagrams for the first time. These are only a high level description and do not show anything approaching the detail of the BUFR tables. There are around 450 BUFR tables, containing around 7000 rows of features, collection elements, codes and flags.

The first 7 diagrams show the BUFR model, while figures 8 to 12 illustrate BUFR coding mechanisms. In ISO terms, these should be separate, but the BUFR standard admixes them to provide what can be considered as “dynamic” models which can define new feature classifications in an instance (a BUFR message).

A simple description of UML connectors is to be found in Annex A and illustrated in Figure A.1. The full definition is maintained by the Object Management Group (OMG) (see reference 6.4).

3.3 BUFR model

Figure 1 shows the main package structure of the WMO BUFR Standard. This structure makes explicit the concepts which are often merged together in the BUFR documentation. It is an attempt to separate format and coding concepts from the data modelling concepts.

Figure 2 describes the structure of the Classification Tables which in ISO terms is the set of BUFR Feature Catalogues.

Figure 3 shows the full set of Table A sub-catalogues. This is the top-level classification of the BUFR catalogue.

Figure 4 lists the BUFR generalised coordinate system. In ISO terms this is a mixture of three functions: the first is a grouping mechanism by which features and coverages are linked, the second describes Coverages and coverage grid definitions and the third can be seen as a way to define feature attributes. Consequently BUFR coordinates are much more variable than ISO coverages. BUFR coordinates also allow an enormous flexibility in defining continuous or discrete coverages.

Figure 5 explores the classification of simple BUFR features. BUFR has no explicit mechanism to define inheritance and association, but in practice the Table D defines sequences and groupings, which serves such functions in conjunction with the BUFR coordinate mechanisms. This is very flexible, but this flexibility also means that it is not possible to enumerate and define all potential BUFR Feature Catalogues.

Figure 6 expands the BUFR simple feature class for temperatureFeatures which are all continuous coverages.

Figure 7 picks on two simple features in the class of observedFeatures, “Type of precipitation” and “Ice development”. These are discrete coverages and show the detailed definition of the discrete values they can take in the BUFR code tables.

3.4 BUFR messages – coding of BUFR instances

Figure 8 illustrates the structure of a BUFR Message. This describes the logical structure at a level close to the physical structure, and is what most BUFR documentation starts with.

Figure 9 is a diagram of the four types elements of BUFR coding operators, one of which, the replication operator is described fully.

Figure 10 lists the BUFR coordinate operators, tightly allied to the coverage classes shown in Figure 3.

Figure 11 discusses the feature descriptor qualifiers. These are operators which perform some of the dynamic feature modification described in class 31 of Figure 4. As well as increased data compression, these operators allow the specification of dynamic coverage grids within a BUFR message or instance.

Figure 12 lists the coding operators of BUFR Table C which explicitly modify the coding “attributes” of any BUFR feature. These are operators which modify the coding, presentation or annotation of items or groups of items in a BUFR instance. They have no affect on the BUFR model directly, but they can add annotation or quality measures to a predefined feature.

4 BUFR UML description

4.1 Figure 1: the model overview.

In this view, some of the BUFR concepts are described in top level packages. Not all the packages are shown here, particularly nested packages, but these are listed to illustrate some of the concepts behind BUFR.

4.1.1 Classification Tables

This package is expanded in Figure 2, and explains the linkages between Tables A (shown in Figure 3), Table D, elements of Table B, code tables and some of the Common Code Tables.

4.1.2 BUFR Coordinates

Table B Coordinate descriptors are BUFR mechanisms to declare coverage grids, and more generally, grouping elements. The linkages in this package are described in Figure 4.

4.1.3 Table Elements

The classes for this package are described in Figure 5. Figure 6 describes one element and the one class table for temperatureFeatures.

4.1.4 Code Tables

Figure 6 describes one element of a continuous coverage class of Table B, and Figure 7 shows two elements of a discrete coverage class of Observed Phenomena. There is no practical way to show all 1200 Table B features, 370 of which have enumerations - code or flag tables, containing around 3000 codes and flag values

4.1.5 Coding Methods

Coordinate operators are described in Figure 10, delayed replication in Figure 11, and general coding operators in Figure 12.

4.1.6 Packing Methods

This refers to the extra packing processing which can be used. It is shown in Figure 12, bit it is not explained in this paper, as this is a coding issue.

4.1.7 BUFR Message

Figure 8 shows the components of a BUFR message – or feature instance in ISO parlance.

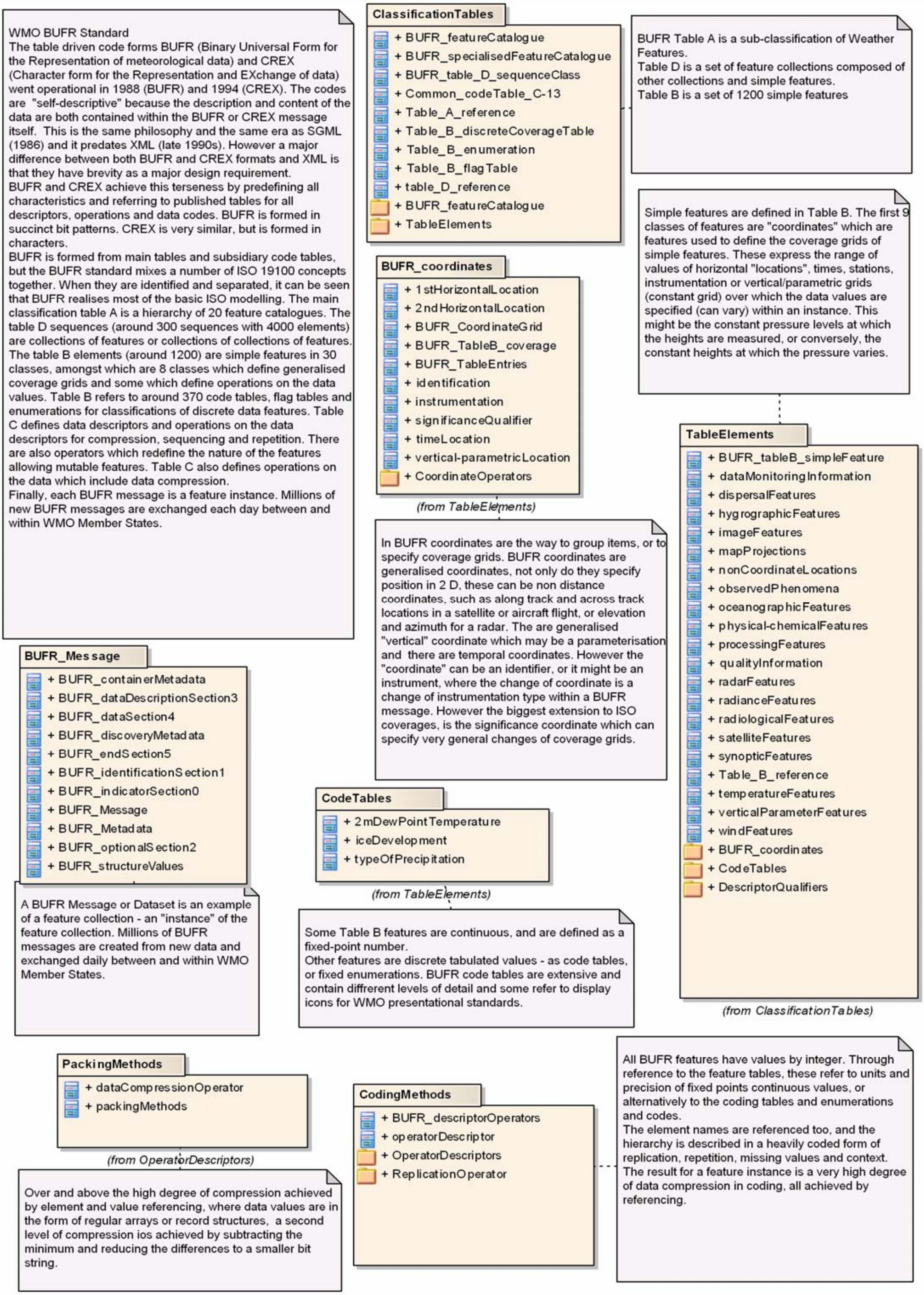


Figure 1: the model overview

4.2 Figure 2: BUFR feature catalogue

This is perhaps the most important diagram of this paper. It summarises the BUFR model and the relationships behind the BUFR tables.

4.2.1 Feature

The ISO 19100 series standards define a "feature" as an "abstraction of a real world phenomenon". BUFR tables are lists of features, and ways in which these features can be further defined. In ISO terms these further definitions can be child features (secondary phenomena), feature attributes (properties of features) and feature associations (relationships between features).

A feature catalogue is a full description of feature types, and a feature instance is an example of a particular feature or feature set. For BUFR, the feature catalogue is equivalent to the BUFR tables and the relationships between the tables, and the feature instances are BUFR messages.

However, the BUFR model allows for features declared in the BUFR tables to be redefined dynamically. This happens in at least 4 distinct ways. The redefinition mechanism is so flexible that it cannot easily be enumerated, and BUFR leaves the definition open to the data producer.

4.2.2 Coverage

In ISO terms a "coverage" is a type of real world phenomena that varies over position. It is expressed as a set of coverage pairs, a position in the spatio-temporal domain, and a record of values, each value in the record associated with an elements measured at the position in an array of points or cells. The geographic paradigm usually associates coverages with regular grids, and features with vector data, but this is too restrictive for the BUFR concept of "coordinates".

ISO adopted the term "coverage" from the Abstract Specification of the Open GIS Consortium, to refer to any data representation that assigns values directly to spatial position.

In meteorology, since nearly everything we measure is a coverage, we need, at the minimum, to extend the domain of the coverage grid from the strictly spatial, to the spatio-temporal and then to the spatio-temporal-parametric domains. A useful rubric is that a feature has an identifier or name (such as a SYNOP, or a tropical cyclone) while the elements in the SYNOP are all coverages such as temperature at the location.

To illustrate:
It is quite common to visualise the cloud amount at a specific height over a regular grid in a region. Here the coverage grid is the horizontal grid, the height value and the time at which it is measured. The coverage is the cloud amount measured at all points in the coverage grid.

But equally, to show a pilot the limits of Visual Flying Rules, we might visualise the height at which $5/8^{ths}$ of cloud occurs at all points in the horizontal grid at that time. In this case the coverage grid is the horizontal grid, the time it was measured, and the point in the parametric coordinate – cloud amount. The coverage is the variable height value.

4.2.3 Limitations of the concept of Coverage for Meteorology.

The question arises – is that extension of Coverage enough for the multi-dimensional atmospheric or oceanographic regime?

By multi-dimensional, we mean more than 3D spatial and time, we intend that the parametric dimensions be included – dimensions in the mathematical sense. In an atmospheric model, at each 4D spatio-temporal location we define many parameters: e.g. temperature, pressure, humidity, 3 winds dimensions, physical-chemical constituents (liquid and solid water, aerosols, perhaps many chemical abundances and rates), fluxes (radiation, heat, fluid) etc..

Many visualisations and data extractions can be derived by projecting this high dimensional model into lower dimensions, using coordinates and CRSs which need not be spatial but parametric (e.g. $5/8^{ths}$ cloud as a coordinate).

ISO 19123 "Geographic information – Schema for coverage geometry and functions" is the ISO/TC211 standard which applies to coverages.

In this coverages are defined as mappings of values to positions, with a defined set of simple operations such as listing and selecting, but also interpolations and evaluations and inversions (mapping from point to value and vice versa with contour production from the grid being offered as an example of an inverse evaluation). The creation of a coverage grid is a distinct task in addition to coverage mappings.

Then much of the definition of the standard is taken up with grid definitions which are relevant to 2 dimensional GIS techniques, where direct interpolation from the raw values is the norm. Hierarchies of discrete and continuous coverages are described which, though it mentions solids and surfaces, is essentially 2D in nature. It has no concept of projections from a multidimensional tensor space.

The BUFR coordinate specification shows that (coverage operations aside) collections, coverages and attribute assignment are all similar functions, which BUFR treats identically as a grouping function.

BUFR only has mechanisms to define coding operations, and to declare operations performed in the creation of the data. It does not define operations which can be performed on the data.

However, rather against the intention described in 2.1, to make the obvious explicit, the whole status of atmospheric modelling is implied. Many of the BUFR "coordinates" have operations dependent on sophisticated atmospheric modelling, rather than only the simple operations defined for ISO coverages.

4.2.4 BUFR feature Catalogue

The top superclass in figure 2 is the BUFR_featureCatalogue. This is the generalisation of the BUFR_specialisedFeatureCatalogues which are realised by the list in table A, expanded with Common_codeTable_C-13. This list is described in Figure 3.

A BUFR_specialisedFeatureCatalogue is composed of BUFR_table_D_sequenceClasses and also of BUFR_CoordinateGrids which is the BUFR way of defining coverage grids.

A BUFR_table_D_sequenceClass is realised by the BUFR Table D. In ISO terms this is a feature collection. A feature collection is also a feature. This recursive property is modelled by composition connecting BUFR_table_D_sequenceClass back to itself. A Table D sequence class can contain other sequence classes.

Ultimately, all BUFR_table_D_sequenceClasses (Table D collections) are composed of Table B elements, BUFR_table_B_simpleFeatures, realised obviously by Table B references. This is modelled in Figure 5 and an example simple feature is shown in Figure 6

However many Table B references are made up of special code tables, if they are not represented by measures of the parameter. Here these are described as Table_B enumerations, discreteCoverageTables and flagTables. BUFR just calls them code or flag tables, and 370 of the 1200 BUFR_table_B_simpleFeatures have code/flag tables. Two examples are shown in Figure 7.

On the BUFR_table_B_simpleFeature class there is a recursive dependency. This shows that some Table B features (more than the coordinate classes) modify other Table B features, and one class of Table B features which does this is class 31 – "Data descriptor qualifier operators".

The BUFR_Table_B_coverage class is a subclass of BUFR_table_B_simpleFeature. Actually the coordinate descriptors are the BUFR classes from 0 to 9 (although 3 and 9 are reserved classes with no members). This is described in more detail in Figure 4.

These are quite complicated to model in this diagram, so the connections shown are necessary simple. BUFR_Table_B_coverages are the features which make up definition of the BUFR_CoordinateGrids. This is shown as being a component part of a BUFR_specialisedFeatureCatalogue, but coordinate grids – coverage grids – can have Table D sequences as values, or be a part of a Table D sequence, defining local grids having Table B simple features as values.

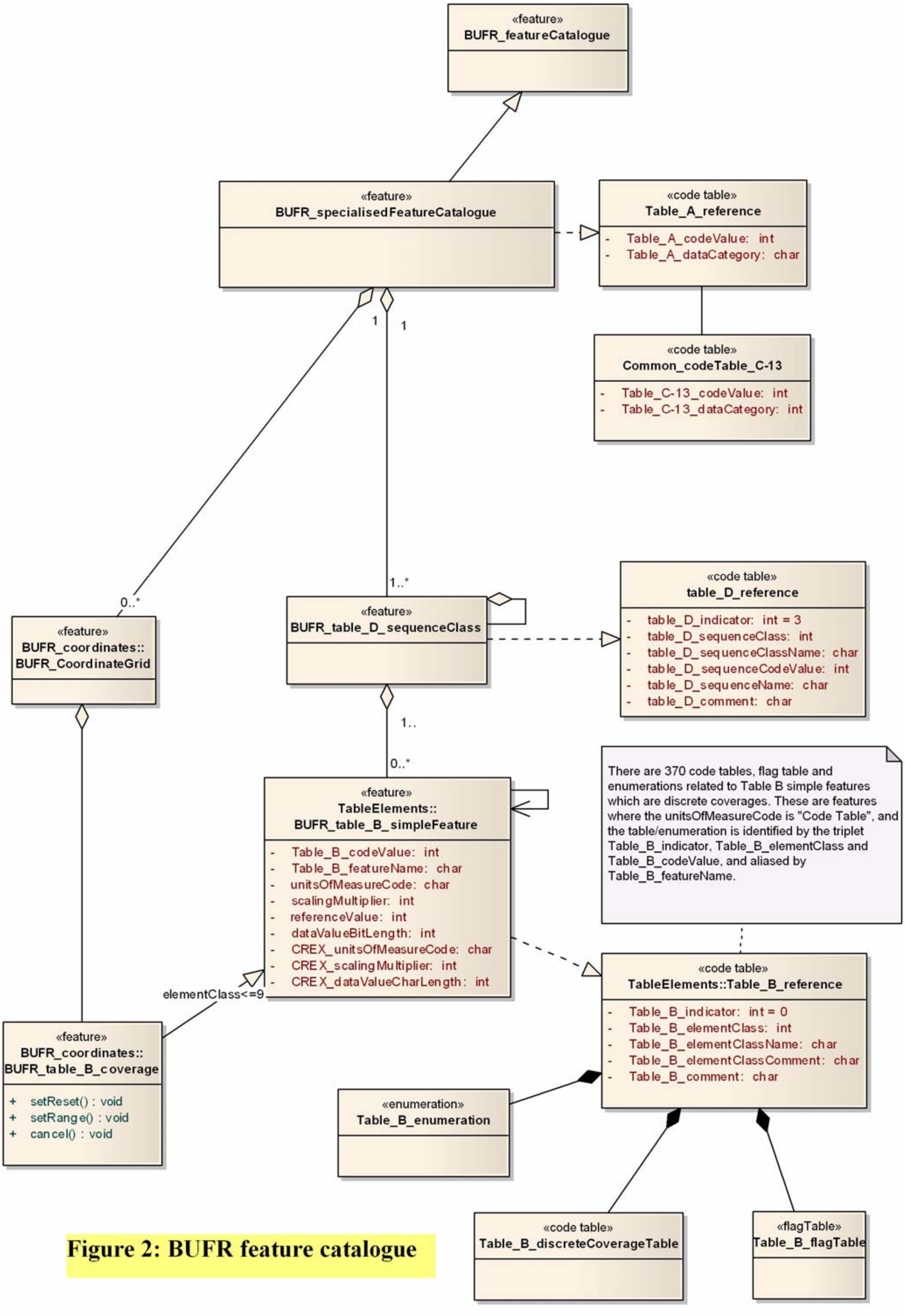


Figure 2: BUFR feature catalogue

4.3 Figure 3: Table A Data Categories

There are 20 sub-classes of Table A which represent sub-catalogues of the BUFR model.

The specialisation of the BUFR_featureCatalogue into BUFR_specialisedFeatureCatalogues is not really a fundamental part of the BUFR model. Many of the component classes of BUFR_table_D_sequenceClasses (Table D collections) and the classes of Table B elements of BUFR_table_B_simpleFeature are used and reused as modules of other feature collections and catalogues. So there are no one-to-one relationships with Table D and B classes and Table B sub-catalogues. There is some attempt to use the classification of Table A catalogues in the classification of Table D, but this works best for some of the more esoteric classes, which do not see much re-use of the component features.

Some of the classifications reflect the way in which the data is gathered, rather than any structural feature. So surface data is split between land and sea (and satellite observations assigned to the surface). Vertical data distinguishes between satellite and standard (non-satellite) soundings, and is further distinguished between single layer data and soundings through the depth of the atmosphere. At least part of this is to recognise that aircraft measurements keep mainly to single layers except at take-off and landing. Part, though is that some satellite products measure parameters at a nominal layer or band, rather than a sounding.

Others are product data: warnings, forecasts, status information, and synoptic features.

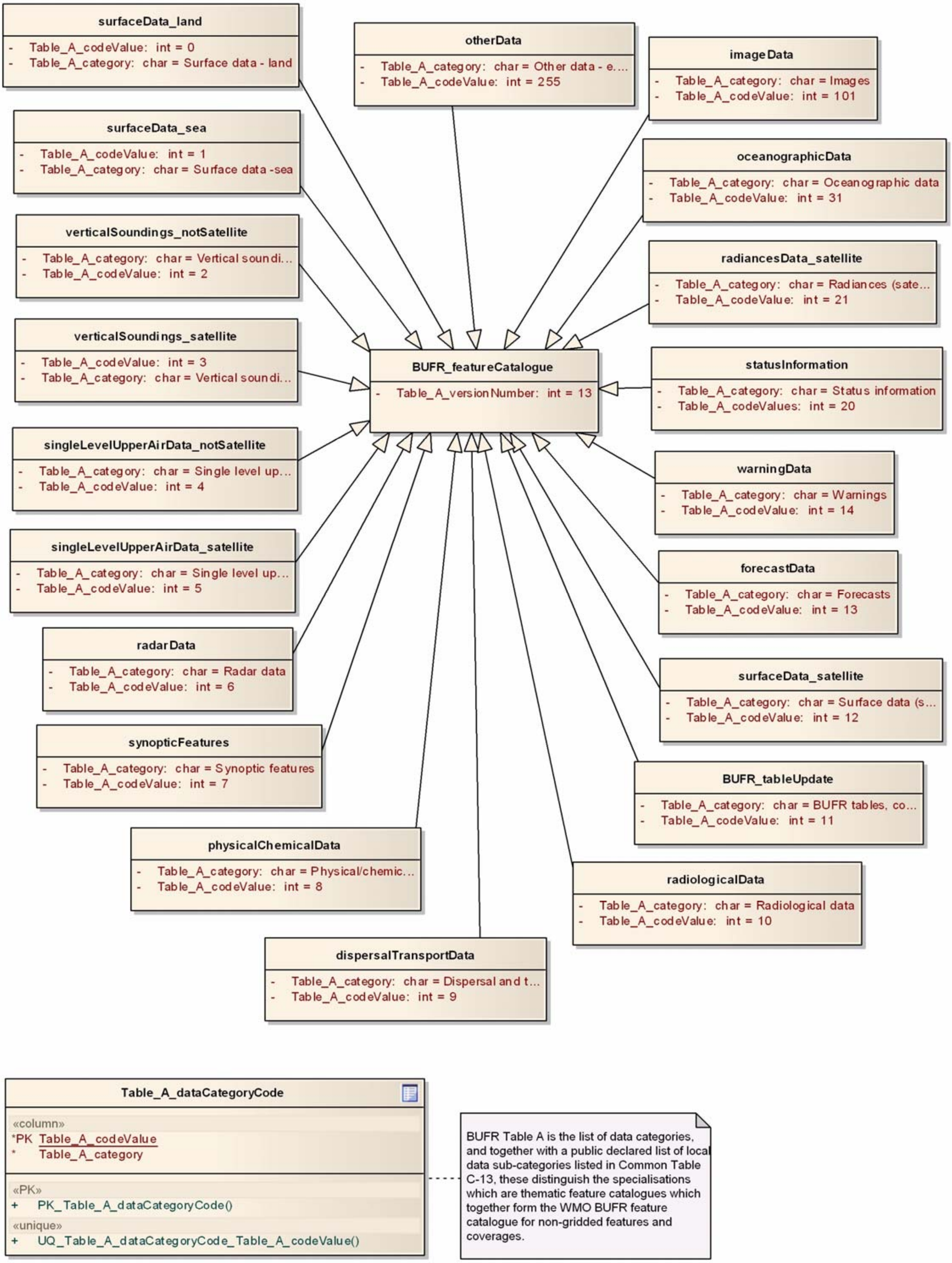


Figure 3: Table A Data Categories

4.4 Figure 4: BUFR coordinates – the BUFR coverage specification

The first three classes of Figure 4 replicate part of the BUFR feature catalogue in Figure 2.

BUFR coordinates aren't restricted to spatial grids. In fact they aren't particularly good for large multidimensional regular grids – GRIB is designed for that. BUFR coordinates are primarily a mechanism both to group sub-features and to define coverage grids, in either case to contain features or feature collections.

The Table_B_references is the realisation of the BUFR_table_B_simpleFeature class. The special code tables (enumerations, discreteCoverageTables and flagTables) also have their use in the BUFR definition of coverages, because the range the coverage grids, particularly for significanceQualifiers is defined there.

The recursive dependency on the BUFR_table_B_simpleFeature class probably does not have a use in the coverage grid, but the functionality (some Table B features modifying other Table B features) is available in the BUFR model – should anyone come up with at use for it.

The BUFR_Table_B_coverage class is a subclass of BUFR_table_B_simpleFeature. While the coordinate descriptors are the BUFR classes from 0 to 9, in practice classes 3 and 9 are reserved and empty classes with no members.

4.4.1 Grouping mechanism to specify BUFR Tables

Since the BUFR tables can be exchanged within a BUFR message, either as complete tables or only updates, it isn't surprising that there is a grouping mechanism (Class 0) for specifying BUFR tables.

4.4.2 Spatio-temporal and "vertical" coordinates

Spatio-temporal coverage grids can be defined with classes 4 to 7.
Class 4 defines temporal location, and it can define absolute values, increments, periods or time displacements, durations or local time displacements.
Classes 5 and 6 are called 1st and 2nd horizontal location. These aren't necessarily distance measures either, but can be angles of elevation and azimuth; rotation and radius; or along and cross track positions for a ship, aeroplane or satellite.

Class 7 is the "vertical" parameterisation, which is not often height, but geopotential, pressure, depth, water pressure.

4.4.3 Identifers and Instrumentation

Class 1 is an "identifier" coordinate. While this is often related to the position of an observing station, it is also the identifier for a moving station, such as an aircraft, or a moving feature, such as a tropical cyclone. Here positions are attributes of the observation, not a BUFR

“coordinate”, coverage or otherwise. In ISO terms, this would be inverted, and the position-record value pairings would be grouped in a “CV_DiscreteCurveCoverage”.
In practical terms, both coding and modelling, BUFR make no such distinction

Class 2 is a “coordinate” describing the instrumentation used to measure the feature values. This is used by the instruments and observing programme of WWW, but is also used in the monitoring and verification programmes. This illustrates another difference in viewpoint from the ISO/TC211 which emphasises position above other attributes of the data, but it also represents a different outlook from other parts of WWW, such as the forecast production programme which emphasises the information in the observation values.

4.4.4 BUFR significance qualifiers

Class 8 illustrates, in the same class, both major similarities and major distinctions between BUFR coordinates and ISO coverages.

Class 8 groups the following data by significance qualifiers, along with Table B class 31 (Figure 11) and other appropriate BUFR coordinates, such as vertical pressure measures, these assign local coverage grids for special observations like a variable length vertical sounding. The Table B delayed replications allows the number of features and replications to be specified in an instance, the significance qualifier will define what sort of level it is, the second coordinate type will assign a value to the level, and the remaining features will specify values of the features at those levels.

This type of qualifier can be regarded, in ISO terms, as a feature attribute mechanism, but this is a dynamic mechanism, and may apply to many feature sets.

However other class 8 significance qualifiers declare the following features to be summary statistics of a data set, in time or space, which is declared in the BUFR instance. Declaring the following features to be first order statistics, difference statistics, or to qualify the values for missing data etc. is effectively a major re-definition of the simple features in Table B.

All but a few of 50 or so sets of class 8 features are realised by code tables specifying attributes of the following set of features. Expanded up to count code table values there are nearly 1000 individual significance qualifiers.

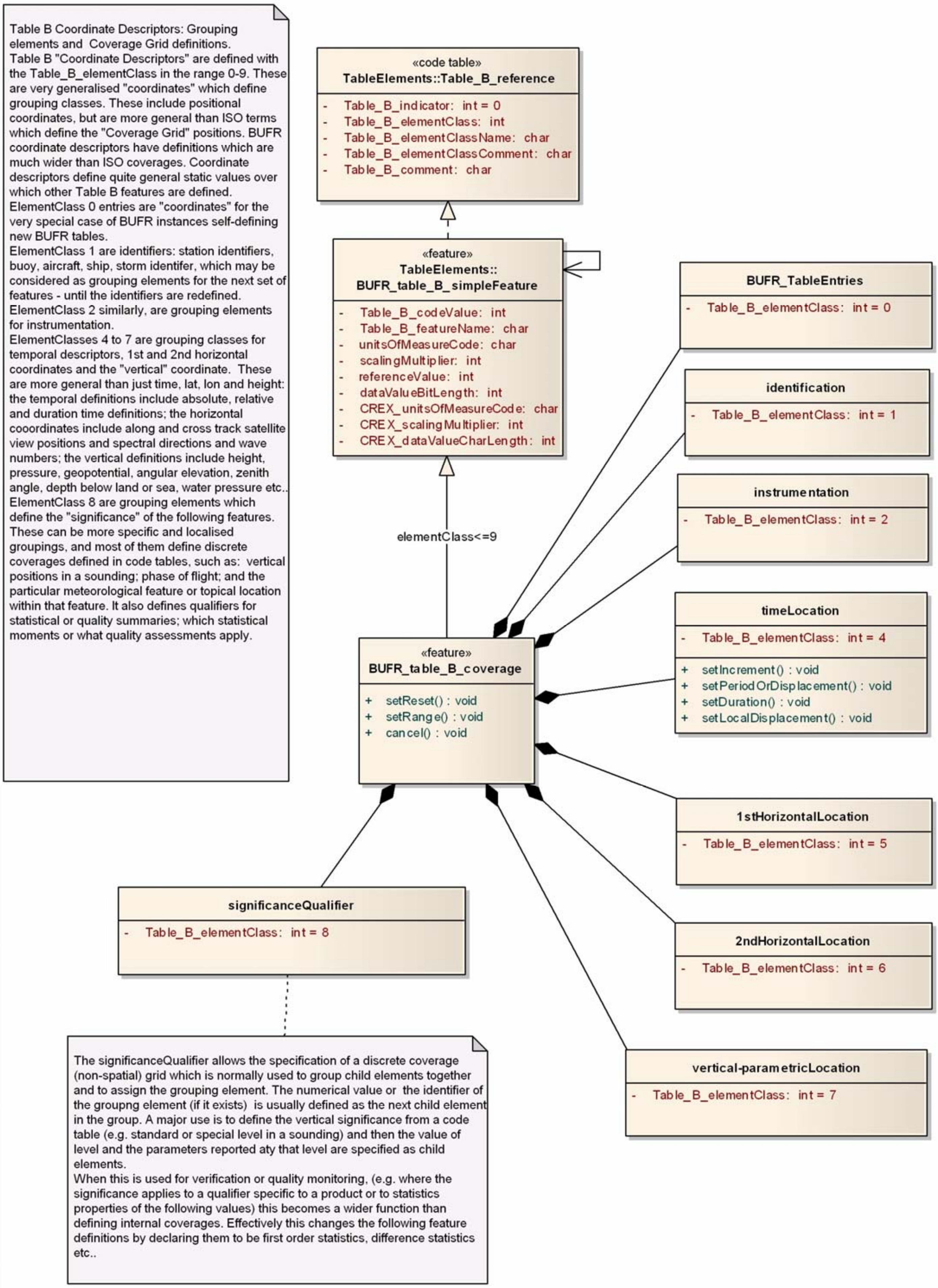


Figure 4: BUFR coordinates - the BUFR coverage grid specification

4.5 Figure 5: Table B simple Features

BUFR_table_B_simpleFeatureTypes are features which are realised by the Table B. Those classes of Table B which are BUFR coordinates are not listed here, as they are modelled in Figure 4.

The feature class is composed of 20 sub-classes of features, some of which reflect the subject classes of Table A and D, others not. The classification of individual features is quite tightly bound to the structure of Table B classes. Since there is no capability for multiple inheritance in BUFR, it is obvious that features which exhibit properties of more than one class have been assigned to a single class.

The recursive association in BUFR_table_B_simpleFeatureTypes represents the class of Table B where features in Table B class 31 operate on other table B features or sets of features. This is shown in more detail in Figure 11.

The temperatureFeatures (Table_B_elementClass = 12) is described in more detail in Figure 6 as an example of a continuous coverage. Two examples of observedPhenomena (Table_B_elementClass = 20) are described in Figure 7.

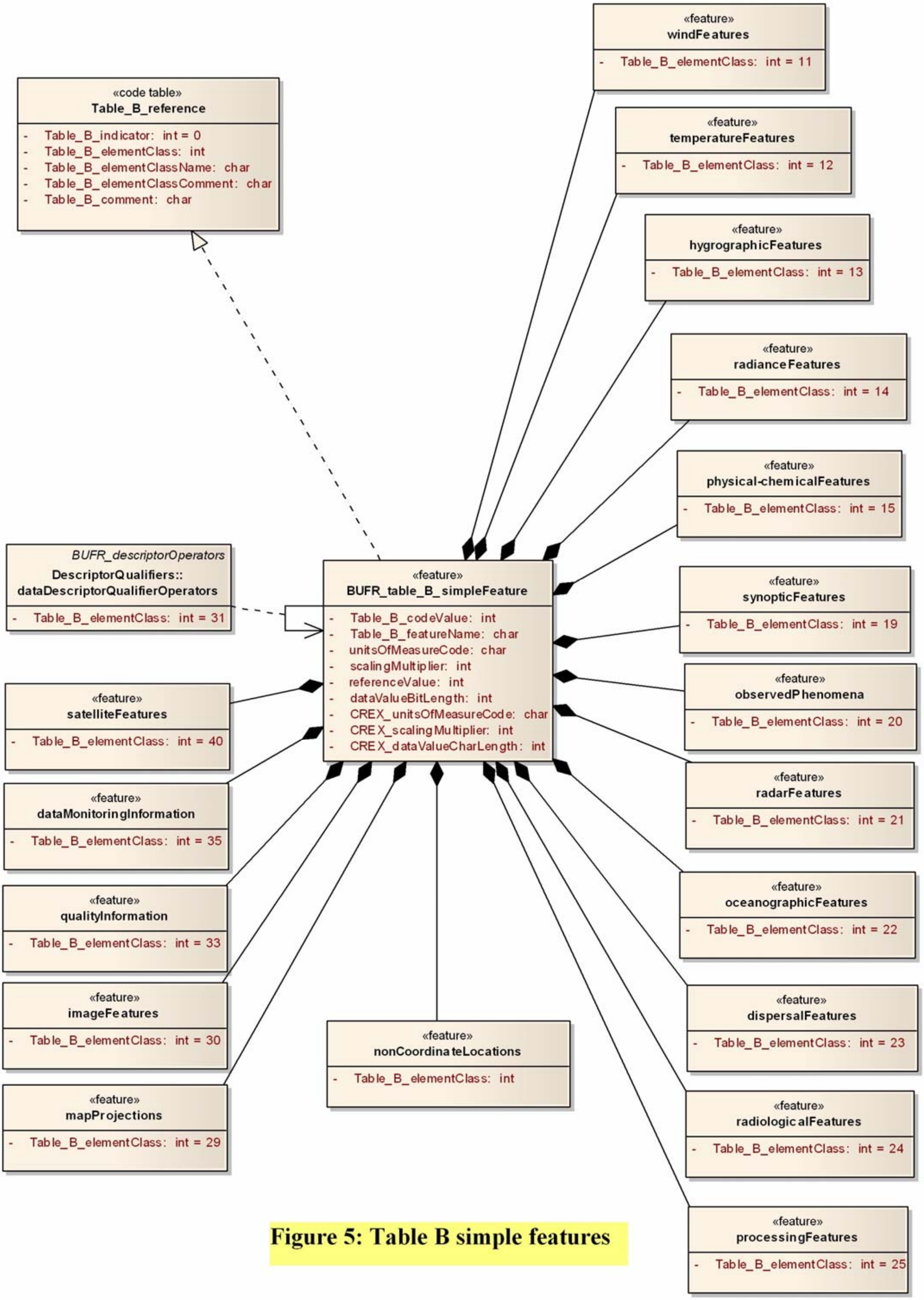


Figure 5: Table B simple features

4.6 Figure 6: Temperature feature code table example of a continuous coverage

The feature class temperatureFeatures is a component of the BUFR_table_B_simpleFeature, and the 2mDewPointTemperature (Table_B_codeValue = 6) is a specialisation of that general class.

Part of the class of temperatureFeatures is listed in Figure 6 as an annotation to the UML. 65 elements (rows) out of 189 are shown.

4.6.1 Number representations in BUFR

This annotation can be used to explain the specification of the number precision mechanism.

The first 3 columns are the expression of the index. The element name makes up the element identifier, and then the BUFR and CREX attributes are listed.

For the Dew-point temperature at 2 m, the units in a BUFR message are expressed as Kelvin. The scale is one, so the value is multiplied by 10^1 before rounding to integer (and so has a precision of 0.1 K), there is no reference offset, and the unsigned integer is stored in 12 bits. This means that the lowest value capable of being assigned to Dew-point temperature and stored in a BUFR message is 0.0 K and the maximum is 409.6 K.
These extremes which can never occur also show how further data compression can work. If the lowest Dew-point temperature in an array of temperatures was 253.0 and the maximum was 303.0, a further reference value of 2330 could be removed from the array, and the range 0 to 500 could be stored in 9 bits per array entry.

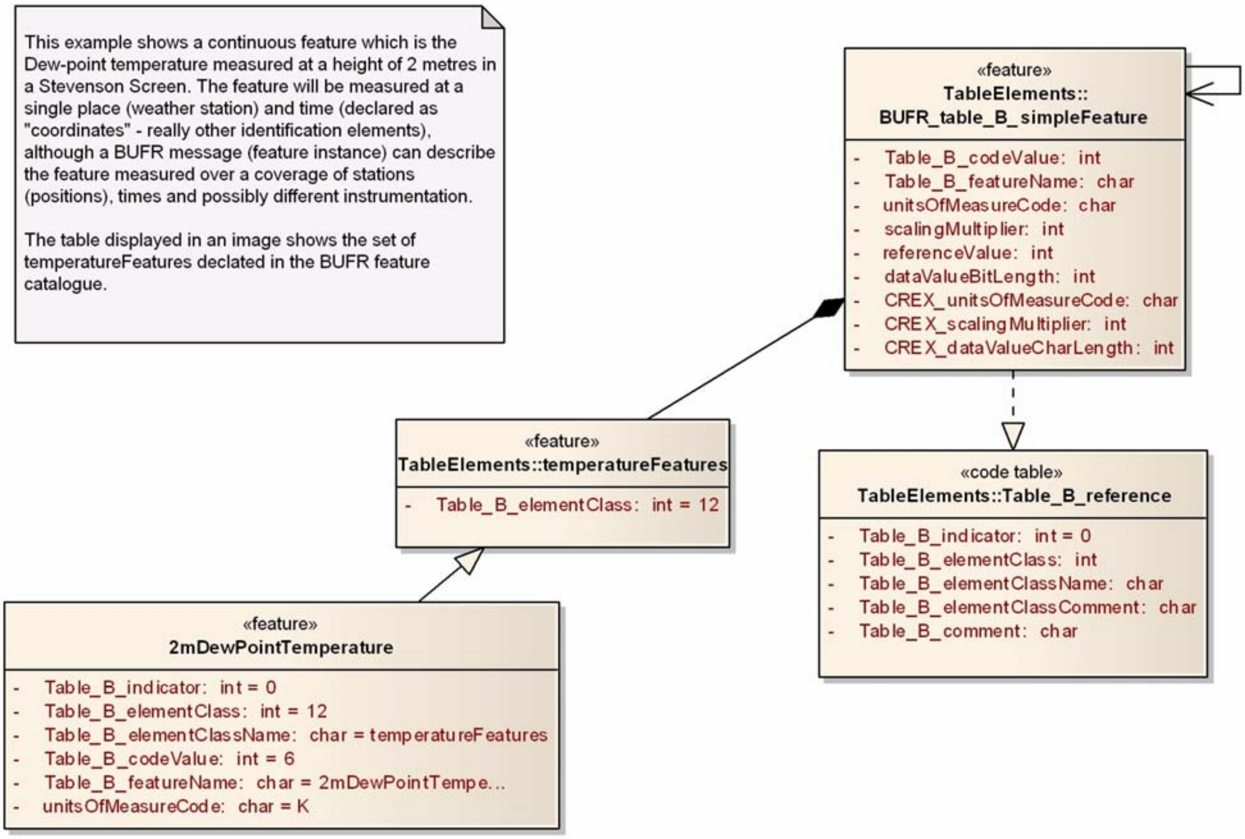
4.6.2 Implicit structures in the temperatureFeature table.

While the data precision for 2mDewPointTemperature (Table_B_codeValue = 6) has a precision of 0.1 K, a higher precision version exists in element Table_B_codeValue = 106. This is stored in 16 bits and has a numeric precision of 0.01 K.

Storing different precisions of the same feature as different elements of the class is a device used by BUFR which ISO would assign as a presentation property, only marginally connected with data modelling.

An ISO feature catalogue might also treat dewPointTemperature (element 3) and 2mDewPointTemperature (element 6) as the same feature with different attributes (element 3 would have a level specification in a coverage grid in the vertical, while the level attribvute for element 6 would be 2metres).
Similarly the minimum and maximum temperatures might be temperatures with specific attributes of maximum or minimum, also at a specific height and over a specific period.

In sub-classing temperature, it might also be thought sensible to model radiation temperatures, skin temperatures and brightness temperatures as specialisations of temperatureFeature which are quite distinct from thermometric temperatures.



				Table_B_elementClass = 12 **Table_B_elementClassName = temperatureFeatures**						
F=Table_B_indicator			**Table_B_featureName**	**UNIT = unitsOfMeasureCode**				**UNIT = CREX_unitsOfMeasureCode**		
X=Table_B_elementClass				**SCALE = scalingMultiplier**				**SCALE = CREX_scalingMultiplier**		
Y=Table_B_codeValue				**REFERENCE VALUE = referenceValue**				**REFERENCE VALUE = CREX_referenceValue**		
				DATA WIDTH (Bits) = dataValueBitLength				**DATA WIDTH (Characters) = CREX_dataValueCharLength**		
TABLE REFERENCE			**TABLE ELEMENT NAME**	**BUFR**				**CREX**		
F	**X**	**Y**		**UNIT**	**SCALE**	**REFERENCE VALUE**	**DATA WIDTH (Bits)**	**UNIT**	**SCALE**	**DATA WIDTH (Characters)**
0	12	001	Temperature/dry-bulb temperature	K	1	0	12	°C	1	3
0	12	002	Wet-bulb temperature	K	1	0	12	°C	1	3
0	12	003	Dew-point temperature	K	1	0	12	°C	1	3
0	12	004	Dry-bulb temperature at 2 m	K	1	0	12	°C	1	3
0	12	005	Wet-bulb temperature at 2 m	K	1	0	12	°C	1	3
0	12	006	Dew-point temperature at 2 m	K	1	0	12	°C	1	3
0	12	007	Virtual temperature	K	1	0	12	°C	1	3
0	12	011	Maximum temperature, at height and over period specified	K	1	0	12	°C	1	3
0	12	012	Minimum temperature, at height and over period specified	K	1	0	12	°C	1	3
0	12	013	Ground minimum temperature, past 12 hours	K	1	0	12	°C	1	3
0	12	014	Maximum temperature at 2 m, past 12 hours	K	1	0	12	°C	1	3
0	12	015	Minimum temperature at 2 m, past 12 hours	K	1	0	12	°C	1	3
0	12	016	Maximum temperature at 2 m, past 24 hours	K	1	0	12	°C	1	3
0	12	017	Minimum temperature at 2 m, past 24 hours	K	1	0	12	°C	1	3
0	12	021	Maximum temperature at 2m	K	2	0	16	°C	2	4
0	12	022	Minimum temperature at 2m	K	2	0	16	°C	2	4
0	12	030	Soil temperature	K	1	0	12	°C	1	3
0	12	049	Temperature change over specified period	K	0	-30	6	°C	0	2
0	12	051	Standard deviation temperature	K	1	0	10	°C	1	3
0	12	052	Highest daily mean temperature	K	1	0	12	°C	1	3
0	12	053	Lowest daily mean temperature	K	1	0	12	°C	1	3
0	12	061	Skin temperature	K	1	0	12	°C	1	3
0	12	062	Equivalent black body temperature	K	1	0	12	°C	1	3
0	12	063	Brightness temperature	K	1	0	12	°C	1	3
0	12	064	Instrument temperature	K	1	0	12	K	1	4
0	12	065	Standard deviation brightness temperature	K	1	0	12	K	1	4

Figure 6: Temperature feature class.
An example of a continuous coverage.

4.7 Figure 7: Observed Phenomenon code table example of a discrete coverage

The feature class observedPhenomena is a component of the BUFR_table_B_simpleFeature with a Table_B_elementClass value of 20. The two features typeOfPrecipitation (which has a Table_B_codeValue = 21) and iceDevelopment (Table_B_codeValue = 37) are both specialisations of the observedPhenomena general class.

The unitsOfMeasure attribute for both features have the value “code table”, so the features are realised by code tables which are referenced by the index of their table B element.

The typeOfPrecipitation element has an index 0-20-21 and codeTable_0-20-21 realises the feature. Actually codeTable_0-20-21 is a flag table with a length of 30 bits. The annotation attached to the flag table shows all the flags. Since at least some of the types of precipitation, whether drizzle, rain, snow grains, diamond dust etc. are considered possible to occur at the same time, the observer can record as many as necessary.

The iceDevelopment element has an index of 0-20-37 and codeTable_0-20-37 realises the feature. The annotation attached to the code table shows the possible values of the code, and the meaning. This feature represents the state of sea ice reported within a ship observation near or within an ice sheet, and the codes reflect all the potential classes that the ship observer may see.

There are 1200 Table B elements and 370 of these have code of flag tables. All the tables are available in a form similar to the annotations at the WMO web site which is available in the references of section 6.

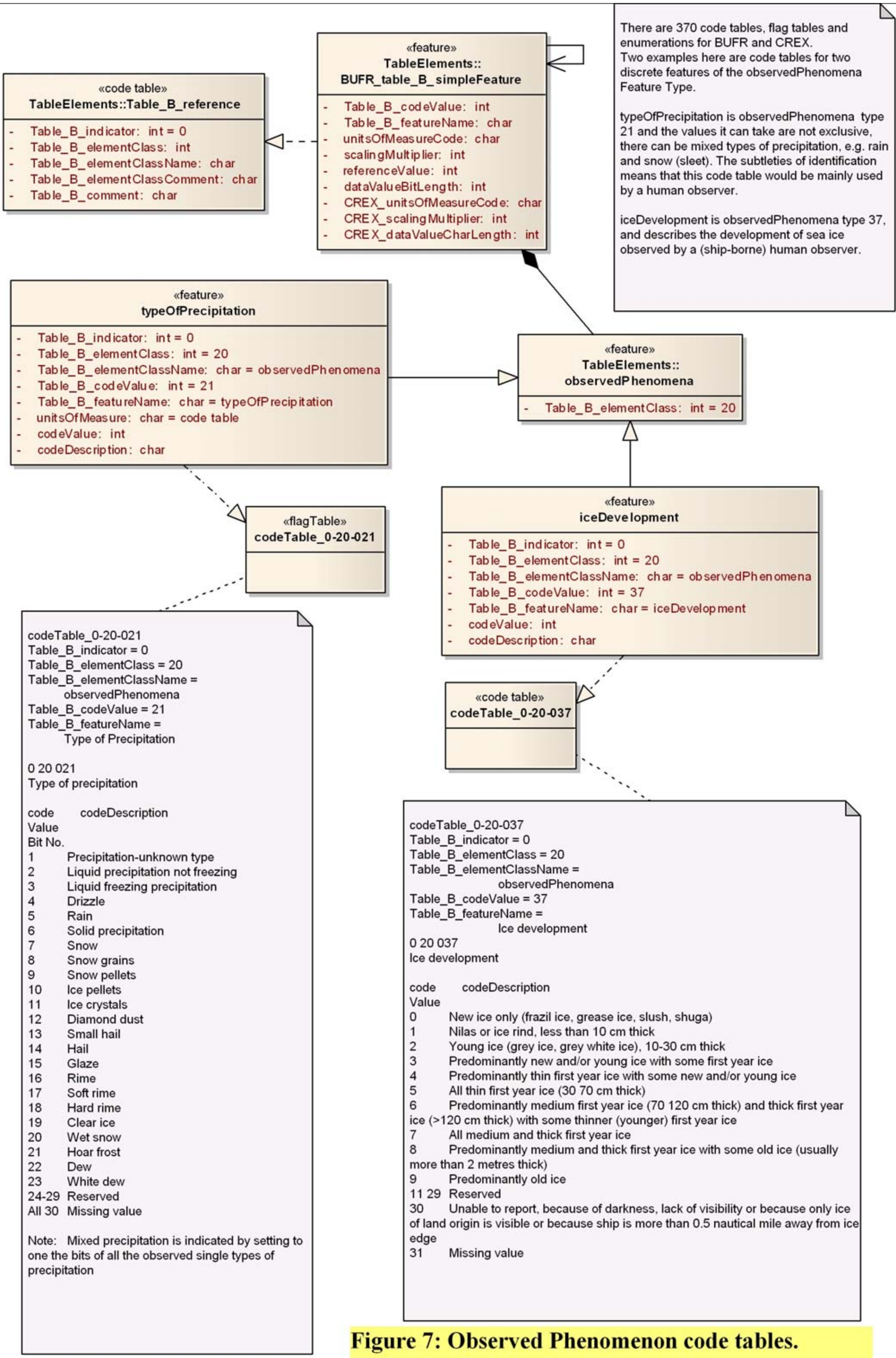


Figure 7: Observed Phenomenon code tables. Two examples of a discrete coverage

4.8 Figure 8: The BUFR message – the feature instance

Figure 8 represents the structure of a BUFR instance – a BUFR message or bulletin sent between WMO data centres.

The structure of a BUFR message is normally what is discussed first in BUFR documentation. This reflects the BUFR designers' view, 20 years ago, that the coding was what was important and what the users wanted to know. The model and the BUFR tables were quite secondary. This paper takes a different view, and considers the BUFR tables, the whole modelling process, and the active maintenance of the standard to be a huge achievement.

Figure 8 is organised slightly differently from other BUFR descriptions of the BUFR code form. These split the code form into 6 sections. Here the code form is considered to have 3 top level components, the BUFR_Metadata, the BUFR_dataDescriptionSection3 and the BUFR_dataSection4.

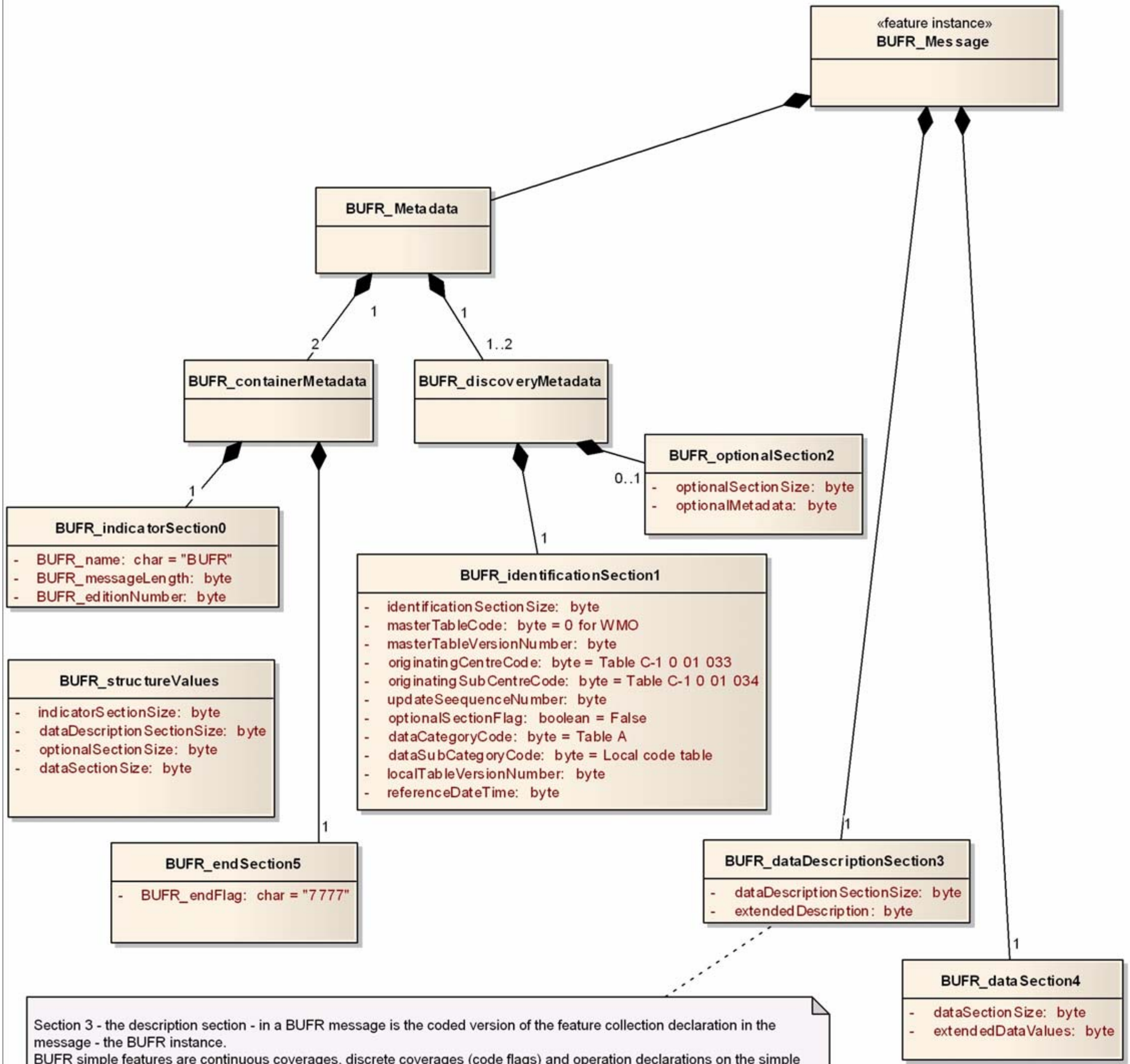
The BUFR_Metadata section contains two sections, BUFR_containerMetadata and BUFR_disoveryMetadata.

The BUFR_containerMetadata is the box structure within which the information is stored. The BUFR_indicatorSection0 has the BUFR identifier, the total length in bytes, and a version number. The BUFR_endSection5 has the end marker "7777". The three values (the identifier, the length and the end marker) together are a check on the integrity of the BUFR box. Each of the BUFR sections has a SectionSize contained in each section. These are all container information, not data, and are listed in the not-connected class BUFR_structureValues.

The BUFR_discoveryMetadata has two component classes, the BUFR_identificationSection1 and the BUFR_optionalSection2.
The BUFR_identificationSection1 has discovery metadata for the data contained within the message. Befitting the BUFR philosophy, these are often references to WMO code tables (Common Code Tables). However, some of what ISO might call metadata is contained within as data.

The BUFR_optionalSection2 is a very interesting addition. Normally it is unused. It is also only ever locally defined, and some repositories which use BUFR as the file form store database indices and foreign key information here as part of the repository structure.

The BUFR_dataDescriptionSection3 and BUFR_dataSection4 store the compressed, packed element identifiers and numeric values and codes respectively.



Section 3 - the description section - in a BUFR message is the coded version of the feature collection declaration in the message - the BUFR instance.
BUFR simple features are continuous coverages, discrete coverages (code flags) and operation declarations on the simple features (data types) which modify those feature types.

Section 3 iis a set of table references to BUFR Table D (for generic feature collections defined for each Table A BUFR specialisation) Table C for operations (such as replications representing meteorological observations at multiple locations and times) on the section 3 references, and references to Table B (simple features and further operations).

When the Table D references are expanded to sets of Table B references, and the Table C operations are performed on the section 3 elements, the result is an expanded set of simple Table B references which give a one-to-one mapping to the values implicit in Section 4.

The elemental Table B references contain codes for units of measure, numerical offsets and multipliers, and the number of bits in which each value is stored in Section 4. There can also further information on any extra compression used on Section 4 values. When this compression is expanded, the bit sections can be separated and the numerical multipliers and offsets applied to the number represented by the bit string to return fixed-point values for features which are continuous coverages. For example this might be a temperature specified to 0, 1 or 2 decimal places in degrees Kelvin.
For discrete coverages, the Table B entries refer to BUFR code tables describing distinct weather elements, such as present weather codes refering to mist, fog, thunder etc..

The BUFR tables define in full detail every non gridded feature known to WMO. The Tables also allow individual instances to be coded extremely concisely (the codes are used for very expensive and very efficient communications from Met Satelites, for example. MSG2 sends out 26GB of new data each day).

Although this seems very different from ISO using GML, the parallels are clear. BUFR Tables are feature catalogues. They are complete descriptions of WMO data and the table mechanism - like ISO feature catalogues - can be extended to many other domains.

Figure 8: The BUFR message – the feature instance

4.9 Figure 9: BUFR descriptor operators

There are 4 types of BUFR operators. These are specialisations of the general class of BUFR_descriptorOperators.

The replicationOperator is a single operator class of its own. It is identified by the indicator (1^{st} level index) having the value 1, and the 2^{nd} level index is the number of terms to be replicated; the 3^{rd} level index is the number of replications.

The coordinateOperator sub-class is the operator implementation in the coding process of the BUFR_table_B_coverage classes of Figure 4. These operators are detailed in Figure 10.

The dataDescriptorQualifierOperators are the operator implementation of the special Table_B class which modifies other Table B features of Figure 5. These are explained in Figure 11.

The operatorDescriptor subclass of BUFR_descriptorOperators are all the classes of Table C. This is shown in Figure 12.

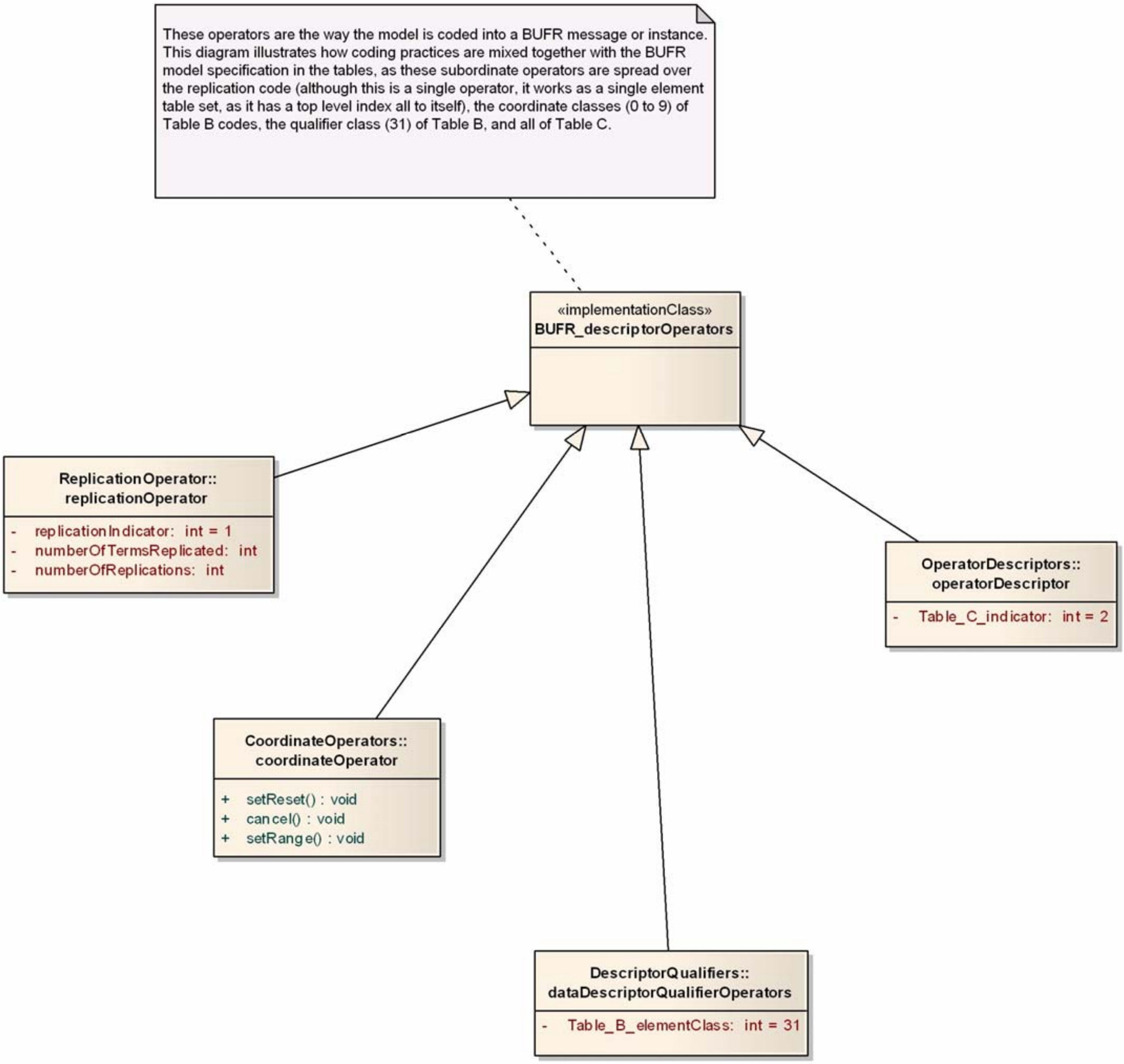


Figure 9: BUFR descriptor operators

4.10 Figure 10: BUFR coordinate operators -coverage specification in an instance

The classes of coordinateOperator are all identified as coverage operators, though the precise function is to group the following terms. This grouping remains in place until it is reset by a following re-definition of the coordinate value. Some of the operators have a cancel function, and others have the option of declaring a range if two operators perform in sequence.

The timeCoverageOperator class can set an absolute time, but also set a time increment, displacement or duration. There are also local displacements which can be set on a restricted set.

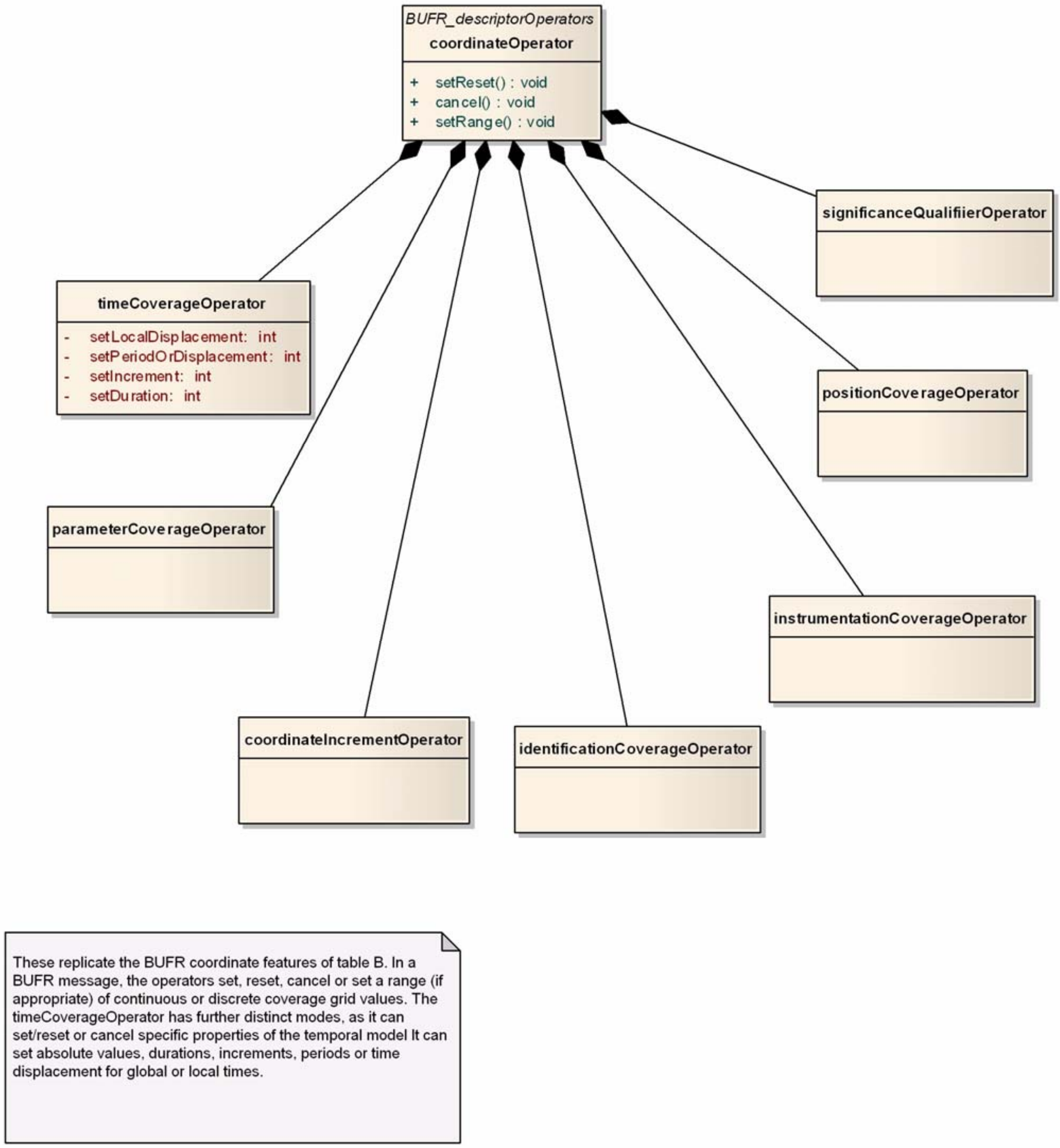


Figure 10: BUFR coordinate operators - coverage specification in an instance

4.11 Figure 11: Operators qualifying the data descriptors

Table B class 31 is the set of dataDescriptorQualifierOperators. This is a sophisticated mechanism to allow a BUFR message or instance to contain variable numbers of elements of one type or a set of types. It is frequently used with the coordinate operators to define number of points in the internal coverage grid.

There are 3 component classes of operator.

The "delayed" mechanism delayedReplicationOperator denotes that the collections or templates do not have a predefined number of terms, so a variable length sounding, or an array of values may follow. While the replication operator modified the descriptor or element numbers, the data sequence will follow the actual number defined in the instance. The delayedRepetitionOperator on the other hand, allows for repeated values of the data equivalent to a run-length packing mechanism.

The dataMissingOperator allows for declaring the data to be present or missing using a single bit in the data value stream, as an alternative to using the full bit length of the element to give the data missing value assuming it is present in the definition of the feature.

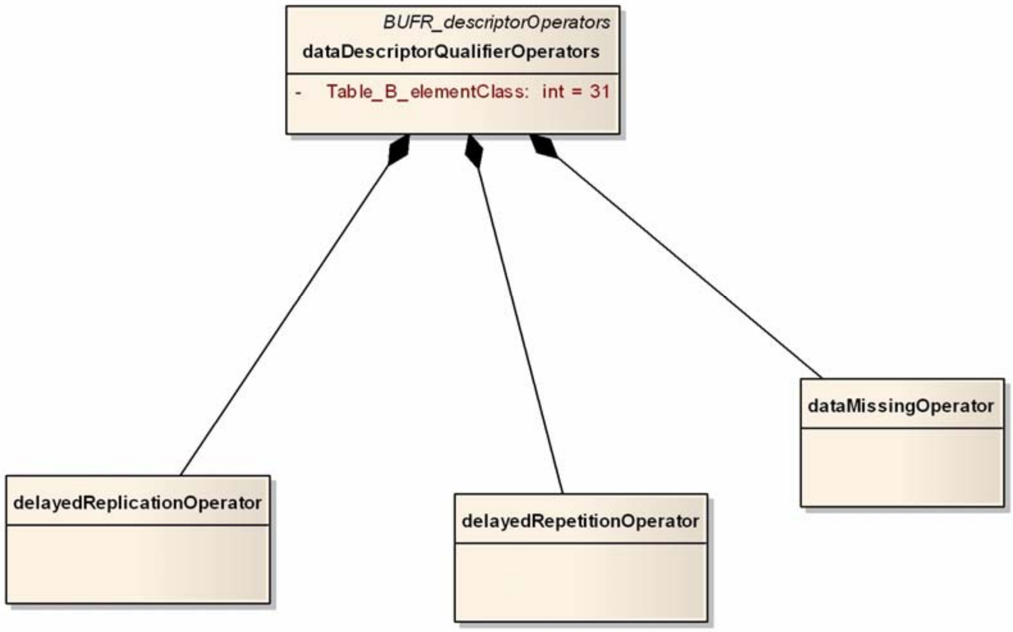


Table B class 31 is the set of data descriptor qualifier operators. These are a sophisticated mechanism to allow a BUFR message or instance to contain variable numbers of elements of one type or a set of types. It is frequently used with the coordinate operators to define number of points in the internal coverage grid. The "delayed" mechanism denotes that the collections or templates do not have a predefined number of terms, so a variable length sounding, or an array of values may follow. While the replication operator modified the descriptor or element numbers, the data sequence will follow the actual number defined in the instance. The delated repetition operator on the other hand, allows for repeated values of the data as a run-length packing mechanism.

The data missing operator allows for declaring the data to be present or missing using a single bit in the data value stream, as an alternative to using the full bit lebgth of the element to give the data missing value (if it is defined).

Figure 11: Operators qualifying the data descriptors

4.12 Figure 12: Coding operators: modifying the attributes or adding annotation.

This demomstrates the class of operatorDescriptors which includes all the Table C descriptors.

The biggest class, and easily the most used of Table C operators are the dataAttributeOperators. The fixed attributes of Table B features explained in figure 6 can be dynamically modified by t6hese operators. The class is composed of the unitsChangeOperator (which modified units-of-measure), the referenceChangeOperator, scaleChangeOperator and the dataWidthOperator. These can be set to apply widely or reset to revert to Table B values.

The packingMethods sub-class applies further compression (as described in 4.6.1).

All these operators have effects entirely on the coding formats. The remaining Table C operators are coding operators, but the add annotation, associated fields or quality assessments which have persistent additions effects the coding. However, there are mostly used for quality assessment, data monitoring and verification, and not to the mainstream Table D sequences.

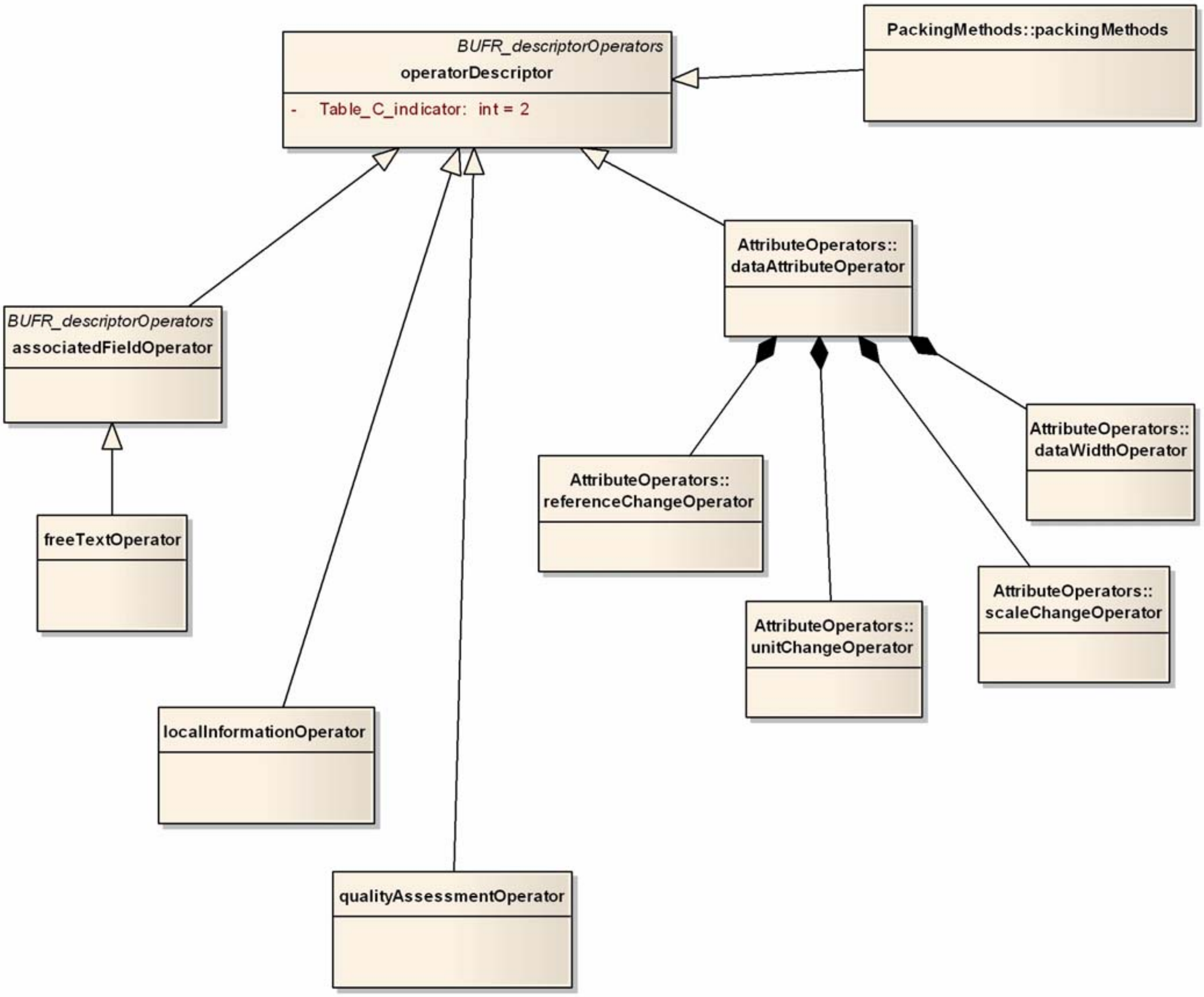


Table C operators are normally for coding a BUFR instance or message. They normally have no reflection on the BUFR model but when they are used to add dynamic attributes or annotations to a predefined template, then the effect of the operations would nee to be reflected in the model, and so to any further code form. However the normal function of a Table C operator is to change the presentation of a feature in an instance by modifying the precision attributes of the units, the scale multiplier, the number of bits, or the reference offset. Table C also controls the dynamic packing methods which are used as an additional compression mechanism.

Figure 12: Coding operators: modifying the attributes or adding annotation.

5 Steps in recasting the BUFR model to derive a GML application schema

Incomplete.
To Be Added

6 References and Code Tables

The first set of hyperlink references are taken from the WMO WWW pages.

6.1 Guide to WMO Table-Driven Code Forms: FM 94 BUFR and FM 95 CREX

Understanding BUFR and CREX.
Layers 1, 2 and 3 *(Note: Layer 3 in English only)* English French Spanish Russian

6.2 WMO Operational Codes -

http://www.wmo.int/pages/prog/www/WMOCodes/OperationalCodes.html

	Operational Codes	Operational on 7 November 2007
BUFR	**BUFR Code Form and Regulations**	Word pdf
	Table A - BUFR. Data Category	Word pdf
	Table B - BUFR/CREX. Classification and definition of data elements	Word pdf
	Definition of Code and Flag Tables associated with Table B - BUFR/CREX	Word pdf
	Table C - BUFR. Data Description Operators	Word pdf
	Table D - BUFR. List of common sequences	Word pdf
CREX	**CREX Code Form and Regulations**	Word pdf
	Table A - CREX. Data Category	Word pdf
	Table B - CREX. Classification and definition of data elements	Word pdf
	Table C - CREX. Data Description Operators	Word pdf
	Table D - CREX. List of common sequences	Word pdf
	CREX Template Examples	Word pdf
COMMON FEATURES:		
	Common Code Tables to BUFR, CREX, GRIB 2 and TAC (Binary and Alphanumeric Codes)	Word pdf
	BUFR/CREX Template Examples, and regulations Templates webpage	
GRIB	FM 92 - **GRIB** Edition 2	Word pdf
	GRIB webpage (to see for GRIB Edition1)	

6.3 For all Traditional Alphanumeric Codes see Manual on Codes

6.4 For the formal UML definition see Object Management Group http://www.uml.org/.

7 Glossary

BUFR	FM 94 BUFR - Binary Universal Form for Representation
CREX	FM 95 CREX - Character form for the Representation and EXchange
CRS	Coordinate Reference System
ET DR&C	Expert Team on Data Representation and Codes
GRIB	FM 92 GRIB - GRIdded Binary
METAR	Aviation Routine METeorological report
OMG	Object Management Group
SGML	Standard Generalized Markup Language
TAC	Traditional Alphanumeric Codes
TAF	Terminal Airfield Forecast
TDCF	Table Driven Code Forms
UML	Unified Modelling Language
WMO	World Meteorological Organisation
XML	eXtensible Markup Language

Annex 1 UML Connections and Relationships

The following has been adapted from ISO documentation. A full description of the ISO conceptual model is in ISO 19103. The formal definition of UML is in the Object Management Group http://www.uml.org/ .

A.1 Associations

An association is used to describe a relationship between two or more classes. UML defines three different types of relationships, called **association**, **aggregation** and **composition**. The three types have different semantics. An ordinary association shall be used to represent a general relationship between two classes. The aggregation and composition associations shall be used to create part-whole relationships between two classes.

The **direction** of an association must be specified. If the direction is not specified, it is assumed to be a two-way association. If one-way associations are intended, the direction of the association can be marked by an arrow at the end of the line.

An **aggregation** association is a relationship between two classes in which one of the classes plays the role of container and the other plays the role of a containee. A **composition** association is a strong aggregation. In a composition association, if a container object is deleted, then all of its containee objects are deleted as well. The composition association shall be used when the objects representing the parts of a container object cannot exist without the container object.

A.2 Generalization

A **generalization** is a relationship between a superclass and the subclasses that may be substituted for it. The superclass is the generalized class, while the subclasses are specified classes. Another common name for a generalisation is an **inheritance**.

A.3 Instantiation/Dependency

A **dependency** relationship shows that the client class depends on the supplier class/interface to provide certain services, such as:

- Client class accesses a value (constant or variable) defined in the supplier class/interface;
- Operations of the client class invoke operations of the supplier class/interface;
- Operations of the client class have signatures whose return class or arguments are instances of the supplier class/interface.

A **realisation** is an instantiated relationship representing the act of substituting actual values for the parameters of a parameterized class or parameterized class utility to create a specialized version of the more general item.

A.4 Roles

If an association is navigable in a particular direction, the model shall supply a “role name” that is appropriate for the role of the target object in relation to the source object. Thus in a two-way association, two role names will be supplied. Different sections of Figure A.1 represent how role names and cardinalities are expressed in UML diagrams.

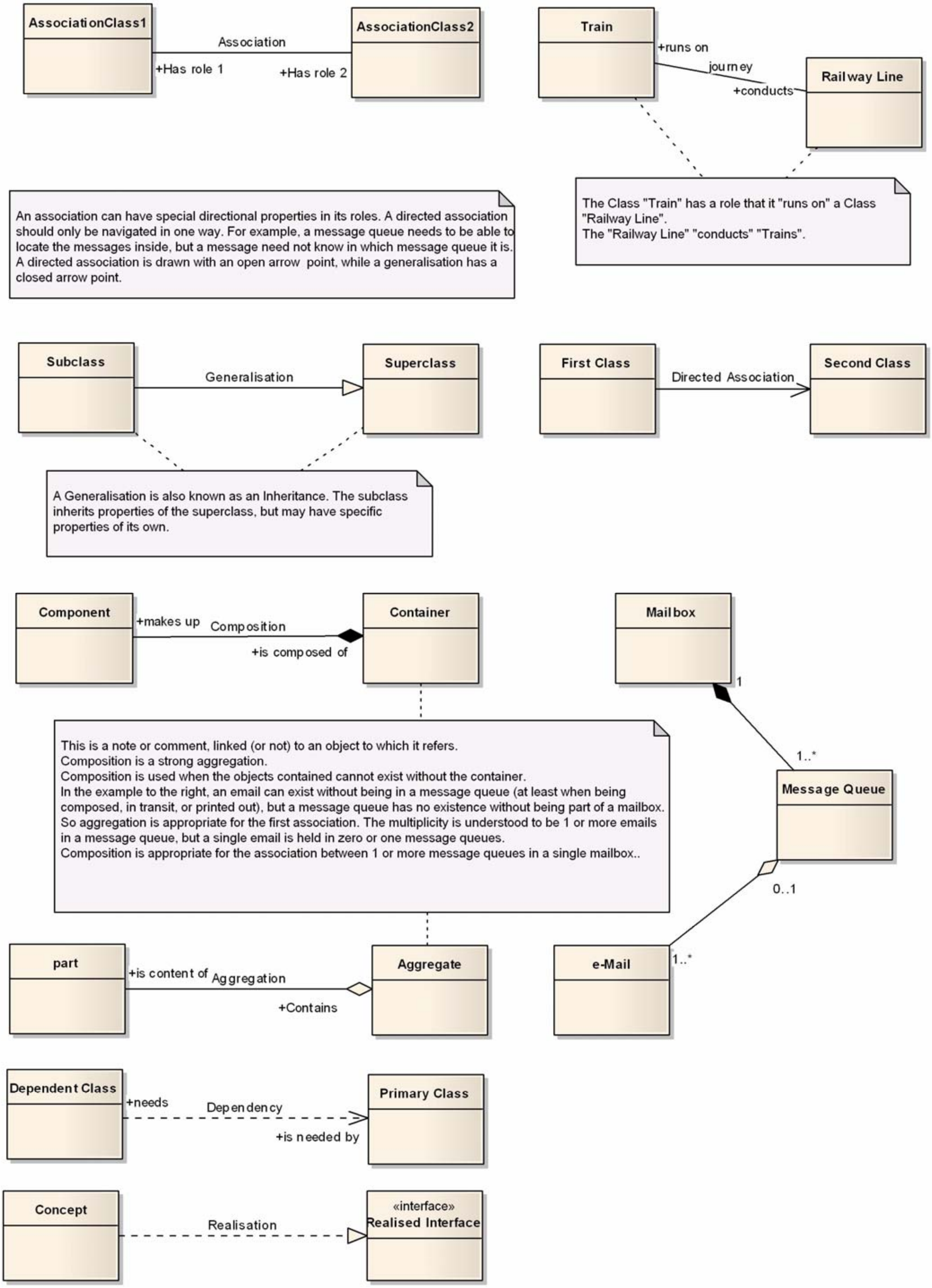


Figure A.1: UML Connections