S-100 Framework Document

Title:	How to build a TC211 conformant Product Specification for Hydrographic Information
Project:	S-100 Work Item 2.6
Source:	IHO TSMAD S-100 Focus Group, Ottawa, April 2007.
Target date:	
Status	Fifth Working draft text
Required action	for discussion on IHO S-100 forum and possible focus group
Reference	S-100-2
File names	
Date last edited	May 8, 2007
Distribution	
Note:	This document contains intellectual property contributed by the UKHO, Canadian Hydrographic Service, NOAA, Seven Cs, CARIS and others, which has been licensed to the International Hydrographic Office as part of their development of a new standard for hydrographic information (probably to be published as S-100). Some of the work is an implementation of international standards published by the Open Geospatial Consortium, DGIWG and/or ISO; those standards themselves contain intellectual property belonging to other organisations, which we believe has been made available for use by publication as open standards.

Comment [PMP1]: Or something like that! Concept to be discussed within the IHO

Contents

1 Introduction

This document describes how to use the various S-100 components together, in order to produce a Product Specification and Application Schema for a hydrographic information product. It includes a generic application schema which is used to create application schemas for specific products. This application schema template does not describe an actual application schema but contains the rules to be used to describe an application schema.

The general approach is:

a) Determine the feature classes properties (attributes, associations) and enumerated values you want - including metadata, quality, coordinate reference system.

b) Check whether acceptable definitions exist in authoritative dictionaries

   - if not, define the new feature classes and properties, and get them registered in the most appropriate dictionary
   - if there is no appropriate dictionary (your feature or property is unique), establish your own.

c) Finalise the product feature catalogue (binding properties to feature classes; documenting allowable ranges).

d) Decide the spatial (geometry), temporal objects and data types you need

   - if you need a geometric or temporal class, or data type, not in the S-100 framework, apply to TSMAD - or abandon thought of being an S-100 family product.

e) Model the feature classes and their associations (create the application schema). You may find you have some more association roles to register.

f) Taking into account the role of GML schema validation in your planned product life cycle, create one or more GML schemas from the logical application schema.

2 References

IHO M-4 Regulations for INT Charts and Chart Specifications

IHO S-44 Standards for Hydrographic Surveys

IHO S-57 Transfer standard for Digital Hydrographic Data.

Harmonised Spatial Profile

ISO 646

ISO 8211 - as used in IHO s-57

ISO/TS 19103:2005 Geographic information – Conceptual schema language

ISO 19107:2003 Geographic information - Spatial schema

ISO 19108:2005 Geographic information - Temporal schema (but only to exclude it!)

ISO 19109:2005 Geographic information - Rules for application schema

ISO 19110:2005 Geographic information - Methodology for feature cataloguing

ISO/DIS 19131 (N1786) Geographic information - Data product specifications

ISO 19115:2005 Geographic information - Metadata

ISO/CD 19115-2 (N1931) Geographic information - Metadata - part 2

ISO 19135:2005 Geographic information - Procedures for item registration

ISO 19136:2007 (N2174) Geographic information - Geography Markup Language

3 Terms, definitions and abbreviations

3.1 *Terms and definitions*

For the purposes of this document, the following terms and definitions apply

geographic information

information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19101]

register

set of files containing identifiers assigned to items with descriptions of the associated items
[adapted from Annex E of the ISO/IEC JTC1 Procedures]

registry

information system on which a register is maintained
[adapted from ISO/IEC 11179-3]

profile
set of one or more base standards or subsets of base standards, and, where applicable, the identification of chosen clauses, classes, options and parameters of those base standards, that are necessary for accomplishing a particular function
[ISO 19106:2004, adapted from ISO/IEC TR 10000-1:1998]

3.2 *Abbreviations*

DGIWG	Defence Geographic Information Working Group
IHB	International Hydrographic Bureau
IHO	International Hydrographic Organization
ISO	International Organisation for Standardisation
TC211	ISO Technical Committee 211 – Geographic Information

3.3 *Notation*

Word comments and *items highlighted & in italics* are still to be resolved by the TSMAD S-100 Working Group.

Comment [PMP2]: To be removed from the final one! Perhaps replaced with something else…

4 General concepts

4.1 *Profiles*

ISO 191xx describes a formal process for creating a profile of a standard. However, in a general sense, a profile is a subset of one or more standards, although it may include community extensions. The IHO S-100 project work items are creating profiles of a range of TC211 standards, but not necessarily document them formally in accordance with ISO 191xx, for example:

Work Item 2.1: Feature Data Dictionary & Registry: ISO 19110, 19126, 19135

Work Item 2.3: ISO 19123, 19129 for Imagery and Gridded Data

Work Item 2.5: ISO 19115 (19130, 19113, 19114, 19139) for Metadata

Work Item 2.6: ISO 19107 and 19111 for Spatial Schema and Referencing

In that sense, this document profiles 19103 Conceptual Schema Language (for simple data types), ISO 19109 Rules for Application Schema and ISO 19131 Data Product Specifications (working from N1786, text for DIS). However, the approach taken is to provide guidelines for creating product specifications and application schemas within the S100 concept.

S-100 conformant data products will use those profiles, as shown in Figure 1 IHO S-100 Work Packages and ISO Standards. *The diagram shows that the IHO S-100 work programme covers all the TC211 aspects, with the exceptions of Quality (perhaps within metadata?), Gazetteer (considered unnecessary) and encoding; remove temporal?.*

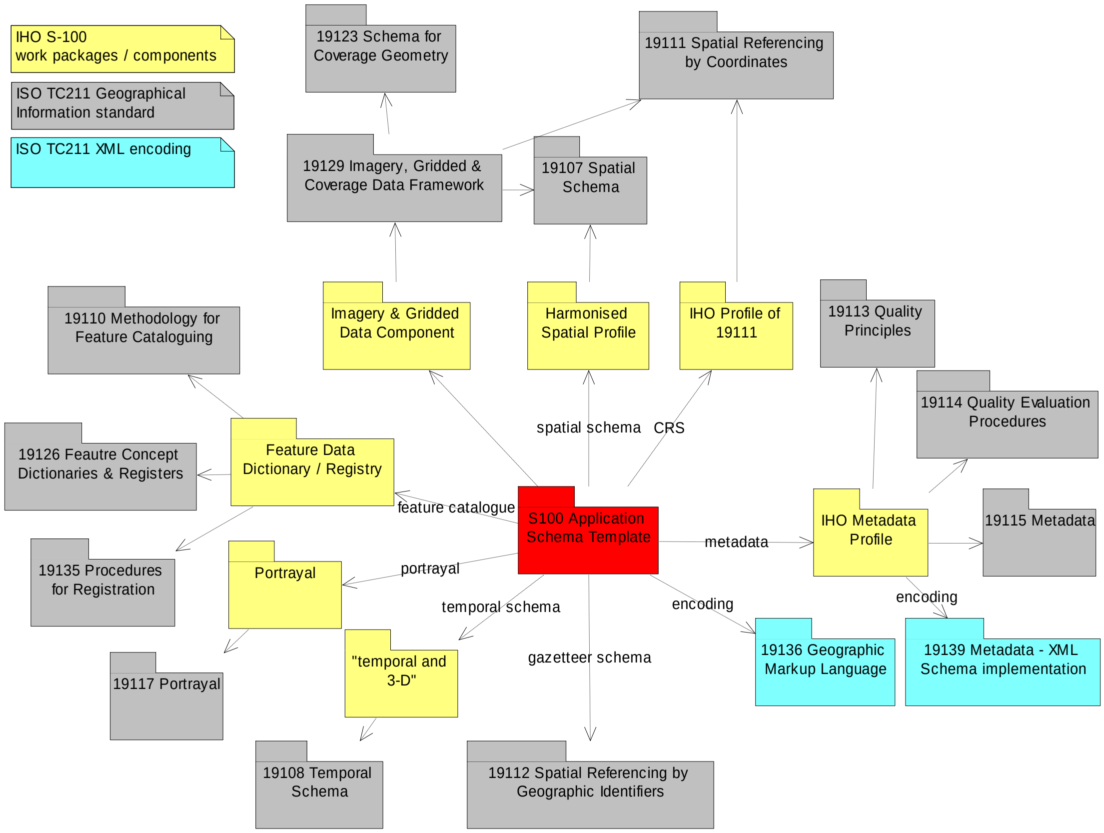


Figure 1 IHO S-100 Work Packages and ISO Standards

4.2 *Schema*

The word “schema” is used in several different ways, all of which are relevant to this discussion:

1. at the broadest level, it encompasses the whole definition of the product - like “scheming” a chart. ISO 19109 says: “An application schema provides the formal description of the data structure and content required by one or more applications. An application schema contains the descriptions of both geographic data and other related data.” It includes the rules for collecting and analysing (cataloguing) the data - for example, it should reference the instructions for hydrographic survey.
2. In a narrower sense, when discussing databases and datasets, schema means the structure of the data - feature classes, properties and the relationships between them.
3. In the specific world of XML and GML, “schema” is often used (in addition to the above) to mean the W3C XML Schema Language Definition, which can be used to check that a given instance (dataset) conforms to the schema (meaning 2).

4 More narrowly, it is occasionally used to mean a single XSD file or a set of XSD files, which represent the schema (meaning 3).

4.3 *Product specification / application schema*

A product specification should give a complete description of a product, including quality and delivery. It includes the application schema, which describes relationships between features, and the feature catalogue, which defines the feature types - by reference to one or more registers. As stated in ISO 19109 "the documentation of the application schema includes a feature catalogue" - from that point of view, the Product Specification is "the documentation of the application schema".

See 5 Product Specification for details.

The Application Schema Template links together the various elements of the ISO geo-spatial standards employed by this standard as shown in .

4.4 *Register / dictionary / catalogue*

ISO 19135 "Procedure for item registration" covers the management of geographic registers. This includes registers of feature instances, registers of feature types, registers of attribute values, registers of attribute definitions etc. Registers of feature types and attribute definitions are often known as data dictionaries. Several registers may be stored in one registry.

ISO 19110 Methodology for Feature Cataloguing requires that all the feature types in a dataset are described in a catalogue; this catalogue can reference any number of definition sources, e.g. registers. An ISO 19109 application schema may use definitions from a feature catalogue, but is free to define further feature types (classes), however ISO 19110 states "A feature catalogue shall be available in electronic form for any set of geographic data that contains features."[1]

The IHO will manage (as per ISO 19135) a compound registry, containing various registers conforming to the same schema. These registers will implement Feature Concept Dictionaries, as described in ISO 19126, such as the IHO Feature Data Dictionary *(the Hydro Register mentioned in "Draft Registry Structure and Procedures Diagram"; currently the "HYDRO" Data Dictionary at the IHO Hydrographic Registry).* This will contain feature types and attribute definitions of navigational significance, referencing definitive official definitions elsewhere, e.g. S-32.

Comment [Peter Par3]: Or is it a hierarchical register? (See NWIP 19126 clause 5.3.2)

Each S-100 product specification shall include a feature catalogue; all the definitions in the catalogue will be sourced from concept or data dictionaries managed as accessible registers. Each product specification / feature catalogue may specify a particular dictionary as the source for all the definitions in the catalogue, or allow a variety.

> For example, an AML application schema could allow data items defined in a DGIWG register in conjunction with some from the IHO Feature Data Dictionary.

S-100 feature catalogues will not be managed as registers[2], in order to emphasise their stability: when an entry in the Feature Data Dictionary (or any referenced register) is revised, or a new definition is added, this doesn't affect the feature catalogue, and therefore doesn't affect the

[1] ISO 19110, Clause 6.1

[2] This doesn't stop the documents themselves being managed *in* a register.

products. A later version of a product specification may choose to adopt the more recent definitions from the Feature Data Dictionary.

4.5 *Feature Catalogue / Application Schema*

ISO 19109 provides rules for converting the information in a Feature Catalogue to be a start for an Application Schema. The Application Schema includes additional information beyond that required for an ISO 19110 Feature Catalogue, in particular spatial referencing, temporal referencing, portrayal. Once spatial and temporal references are included in the Application Schema, this can be expressed in UML with sufficient detail to allow automatic generation of a GML application schema. ISO 19136 provides rules for this, and there are software tools available – which depend on specific UML modelling tools[3].

Alternatively, a 19110 Feature Catalogue could be generated automatically from a 19109 Application Schema. Separate transformations could be created to take constraint information from the Application Schema to assist with dataset validation.

As a third option, one could create a "Feature Catalogue plus", which contains all the information from which Feature Catalogue, Application Schema, constraint rules and GML Schema definitions can be created. *Tenet will share their information model for this.*

To do: decide which approach to take; this document currently follows the 19109 approach.

Which ever way, the Feature Catalogue is "data" which can be loaded by production and user software to understand the datasets, and to perform a certain amount of validation. GML schemas can be used for a certain amount of dataset validation. The feature and attribute definitions, referenced from the dictionaries, can be presented to the users.

Note: the master dictionary definition remains in the IHO (and other) registry; the master feature catalogue for the product remains on the IHO's (or specifying organisation's) web site. Software will often want a local copy of the feature catalogue and dictionary extracts. One way to manage this is by using URN references, with different URN resolvers for on-line and off-line use.

Note: GML application schemas are written in XML Schema Definition Language. This is capable of expressing simple constraints, e.g. minimum and maximum values, character patterns. It is not capable of directly expressing constraints which involve more than one property type (e.g "if there is more than one value of 'colour', 'colour pattern' must be set"). If these are included in the Application Schema, perhaps in a formal language such as Object Constraint Language, the ISO 19136 rules ignore them. Thus the GML schema associated with a given product can only be used for a limited validation.

5 Product Specification

This section to be revised after component S-100-13 Data Product Specifications is complete.

This section draws on ISO 19131 Data Product Specifications. A data product specification describes how the product should be, whilst the metadata accompanying a particular dataset describes how it actually is. That's why many of the sections here overlap with those in ISO 19115 and the IHO Metadata Profile.

[3] Interactive Instrument's "ShapeChange" is available under GPL and commercially, and supports UML models exported from either Rational Rose and Enterprise Architect (& perhaps others).

A Product Specification contains:[4]

a) Overview
b) Specification scopes
c) Data product identification
d) Data content and structure
e) Reference Systems
f) Data Quality
g) Data product delivery
h) Metadata

And optionally…

i) Data capture
j) Data maintenance
k) Portrayal
l) Additional information

A particular product specification and application schema will specify which version of S-100 and its component profiles it relates to.

5.1 *Overview (ISO 19131 Clause 7)*

The overview documents the creation of the product specification and the name of the data product (with any acronyms), and contains an informal description. This should include the purpose of the data (e.g. for navigation, for situational awareness) and may comment on the expected data sources, production processes and maintenance regime.

5.2 *Specification scopes (ISO 19131 Clause 8)*

This section is only used where different parts of the product (e.g. by theme or geographical extent) have different specifications. For example, some aspects of the specification may be specific to bathymetry, or to non-tidal waters. If this is the case for the product being specified, this section defines the various "scopes" within the overall product specification, and how (e.g. by code) they should be identified in the datasets.

5.3 *Data product identification (ISO 19131 Clause 9)*

This section describes how to identify data sets that conform to the specification – title, version numbering conventions, key words to be used. See also Metadata / Dataset identification.

5.4 *Data content and structure (ISO 19131 Clause 10)*

This section is the Application Schema, which "defines (the) content and structure of data"[5]. See 6 Application Schema (19109 Clause 7) (19109 Clause 7) for further details. The documentation of an S-100 application schema shall include a Feature Catalogue. Allowable units of measure will be defined by reference to a UoM dictionary; producing organisations by reference to a suitable dictionary/register.

---

[4] ISO 19131, clause 6
[5] ISO 19109

If the application schema is a separate document, then the product specification shall include a narrative summary.

5.5 *Reference Systems (ISO 19131 Clause 11)*

The product specification declares the coordinate reference system (or choice of coordinate references systems) available for products of the type being defined. The application schema will show how CRS references are carried in the data sets; this may be by reference to a CRS dictionary. The application schema will also define the spatial data types allowed. See also Metadata / Reference Systems; it is likely that the specific reference systems used in a dataset will also be carried in the encoded dataset.

5.6 *Data Quality (ISO 19131 Clause 12, and see ISO 19113)*

Each product specification will describe the data quality requirements. One aspect is the “data quality overview element” which should allow a user to decide whether this dataset is the one they want. The other aspect is the metadata allowed for specific feature collections, features and attributes within the dataset.

The data quality overview element should include at least the intended purpose and statement of quality or lineage.

Other data quality elements cover: completeness, logical consistency, positional accuracy, temporal accuracy, thematic accuracy, and anything specifically required for the product being specified. The product specification should comment on which of these are to be used and how, including a description (or reference to) conformance tests. For example, should data only be published if it passes a particular test, or is it allowable to publish the data with a quality statement which indicates non-conformance.

The product specification shall describe how each quality element is to be populated, for example stating the mechanism to reference the quality evaluation procedure, and allowable values for the quality results.

The application schema shall indicate how the data quality elements will be related to the data items, for example whether a particular dataset should have homogeneous quality, or whether quality elements can be related to feature collections, individual feature objects or attributes. Finally, the encoding description (see 1.17) shall indicate how the quality elements will be encoded.

See also S-100 Component 5-1 Metadata for dataset quality, and S-100 Component 5-2 for detailed Data Quality.

Comment [Peter Par4]: Cross check / revise after completion of Meatadata part 2.

5.7 *Data capture (ISO 19131 Clause 13)*

This section is optional in ISO19131, but is strongly recommended, at least for IHO navigational products.

The product specification includes the collection criteria for mapping real world objects on to the conceptual objects of the dataset. This may be done in the definition sources for the particular objects, but the product designer may want to constrain this further (“only include depths surveyed to…”).

See also IHO Publications S-44 and M-4.

Data products can carry information about their data sources (metadata lineage elements); the product specification and application schema will show whether this is expected, and how it is to be done.

5.8 *Data maintenance (ISO 19131 Clause 14)*

This section is optional in ISO19131, but is strongly recommended, at least for IHO navigational products.

It should describe the principles and criteria applied in maintenance decisions, as well as expected frequency of updates.

See also Metadata / Maintenance Information.

5.9 *Portrayal (ISO 19131 Clause 15)*

This section is optional in ISO19131, but is strongly recommended, at least for IHO navigational products.

Classes and attributes required to support portrayal for a particular product need to be registered in a Feature Data Dictionary and the Feature Catalogue for that product specification. Examples could be cartographic object classes, scale maximum / minimum attributes, hinting attributes of textual information objects.

See the output of the proposed joint TSMAD/Colours & Symbols work item (TSMAD Work Item 2.8) on portrayal (including revision of S-52) for further details.

Possibly see also Metadata / Portrayal Catalogue Reference

5.10 *Data product delivery (ISO 19131 Clause 16)*

This section includes a description of file structures and format, as well as medium. The file structure (encoding) could be specified completely here, or by reference to a separate profile or standard. S-100 gives guidance on GML (ISO 19136) encoding. A given product would have a specific GML application schema, expressed in one or more XML Schema Definition Language files. *To do: include GML examples.*

S-100 GML should use strong typing, preferably using a model driven architecture. The definitions in the Feature Data Dictionary should support this, by including “camel case” names. *The IHO will work to provide a toolset to support creating an application schema from the definitions in the FDD, and to generate GML schema definition files from the application schema.*

Specialised products may use other encodings, for example S-100 contains a profile of ISO8211 binary encoding, intended for use in the backward compatible S-101 product specification.

See also Metadata / Distribution

Comment [PP5]: NOAA interested in "HDF5" for Open Navigation Surface BAG(which is in the IGD component)

Comment [Peter Par6]: Should we mention HDF5, which I'm told is used for Open Navigation Surface, which comes under BAG in the IGD component?

Comment [Peter Par7]: I would not expect this to be with the dataset metadata, but would expect to find it in catalogue metadata.

5.11 *Additional information (ISO 19131 Clause 17)*

Anything else, for example constraints on access and use; details of recommended training; related products.

5.12 *Metadata (ISO 19131 Clause 18)*

This section describes the metadata, by reference to one or more metadata dictionaries, which will encapsulate the IHO profile of ISO 19115 and 19115-2. This may include reference to a data quality dictionary/register, defining evaluation methods. The application schema will show how metadata is carried in the datasets.

6 Application Schema (19109 Clause 7)

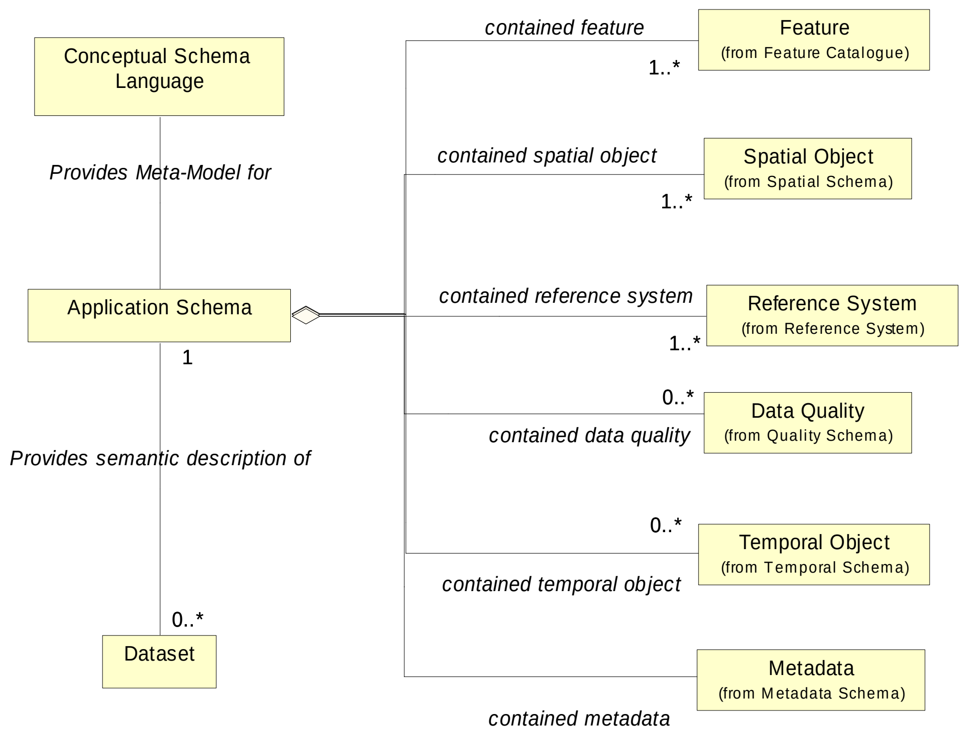


Figure 2 Application Schema parts

Note: the application schema in this diagram doesn't include portrayal, as it isn't passed in the dataset – it will be defined in the product specification. This diagram doesn't show the links which exist between the Metadata and Data Quality components, or between the Metadata and Reference System components.

Comment [Peter Par8]: Is this true?

An S-100 Application Schema is created by first converting the feature types and property types from the Feature Catalogue into a UML model, following the rules laid down in ISO 19109. The following steps cover those rules, and the encoding rules from ISO 19136, and only cover things things which are included in S-100[6]:

1 Feature types become UML classes stereotyped <<FeatureType>>, and named using the camelCaseIdentifier of the FeatureType in the Feature Data Dictionary[7]. *To decide: do the camelCaseIdentifiers appear in the Feature Catalogue?*

2 Information types become classes with no stereotype, named using their camel case identifier[8].

3 Complex attribute types become classes stereotyped <<DataType[9]>>, again named using their camel case identifier. Associations from the feature and information types which use the complex attribute types shall be compositions (implying the value of the attribute has no existence if no objects are associated to it). [8]

4 Simple attribute types become attributes of the relevant class

  4.1 multiplicities other than “1” shall be specified explicitly

  4.2 The UML attribute takes the type from the Feature Catalogue *(except FormattedText, and assuming we rename “text” to CharacterString)*

  4.3 *The itemIdentifier from the Feature Data Dictionary should appear as an attribute of the feature type, in order that the definition and remarks can be accessed.*

  4.4 Unique identifiers according to XML’s ID mechanism are used to identify feature and information objects. This only guarantees uniqueness within the dataset. The Product Specification shall indicate if this is to be used for a domain-wide persistent identifier.

  Note: the ISO 19136 rules do not deal with constraints, even simple lower and upper bounds which could be implemented in GML schema.

5 Enumerated attribute types become classes stereotyped <<Enumeration>>, again named using their camel case identifier. Each enumerated value becomes an attribute of that class.

6 Feature to feature associations become association between the relevant classes, with the associated roles implemented as role names. *To do: consider navigability of associations, and whether we need association classes to handle many to many associations.*

  6.1 aggregation roles become aggregations

  6.2 composition roles become compositions

  6.3 simple association roles become role names

Having translated the Feature Catalogue, the Application Schema is completed by adding the spatial properties[10] and any portrayal elements. This can then serve several purposes:

---

[6] *I have used the names from the Feature Data Dictionary, until the Feature Catalogue model settles.*

[7] If using a feature type from a data dictionary which doesn’t provide camel case names, these will need to be introduced in the Product Specification.

[8] Extending from 19109, with reference to 19136

[9] *Or “Type”: I’m not clear from 19136*

[10] These seem good candidates for inclusion in the Feature Catalogue, even though 19109 assumes they aren’t.

- UML class diagrams from the Application Schema can illustrate aspects of the Product Specification.
- It can be converted to a GML application schema, using the rules specified in ISO 19136, or an automatic tool.
- It may be used to build database schemas or software implementation classes.

This document shows how the classes defined in the various S-100 work packages could be combined to create specific application schemas for specific products. This document only gives indicative examples for various basic configuration which product specifications could use as the basis for a comprehensive application schema. *S-100 could include an "empty" UML model, which contains the geometric and other basic components this is the approach taken by "HollowWorld[11]".* Feature classes in these examples (mostly abstract super types) only include a selection of the attributes normally required.

Example: S-101 will be the new ENC Product Specification. A further S-100 application schema could exist for pilot services. Being within the S-100 family should enable those pilot service datasets to be integrated, or to interoperate, with S-101, for example in a bridge system.

This section discusses the matters which need to be considered when developing an S-100 application schema, with references to ISO 19109 clauses. See for the reasons why some parts of ISO 19109 are not used.

As this is a generic application schema a wider scope is used than may be required by specific products. Additional constraints may be added in the individual product application schema.

6.1 *Features (19109 Clause 7.1)*

For S-100 application schemas, the features to be used will be defined in a Feature Catalogue, drawing definitions from one or more registers, primarily that to be established by the IHO.

[11] https://www.seegrid.csiro.au/twiki/bin/view/AppSchemas/HollowWorld on Australia's Solid Earth & Environment Grid Application Schemas TWiki.

6.2 *Features and the application schema (19109 Clause 7.2)*

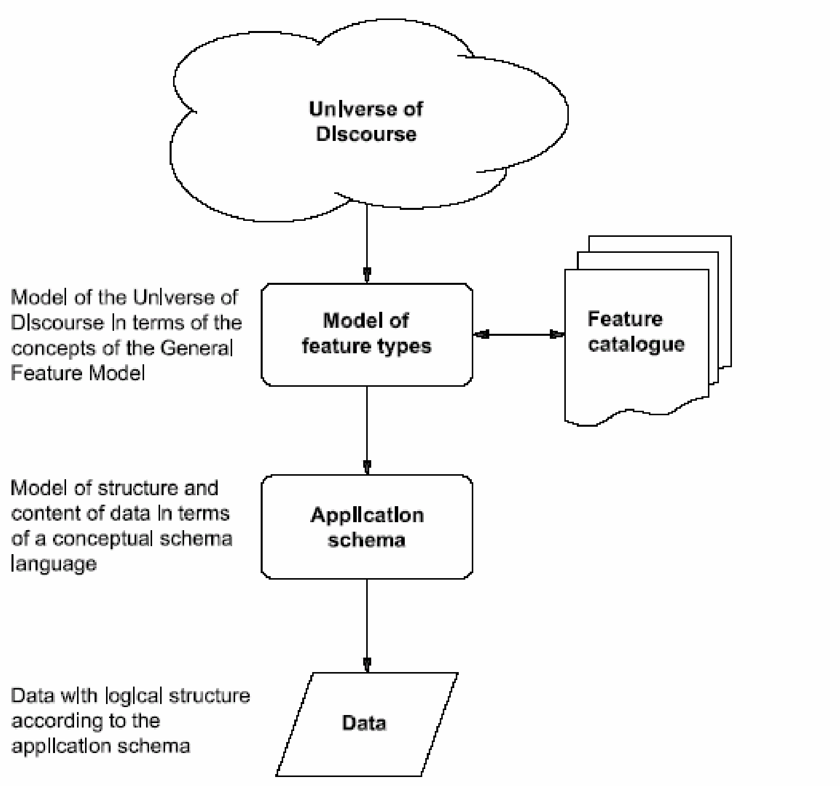


Figure 3 From reality to geographic data[12]

The product specification includes a feature catalogue, which specifies the feature types which can be used, by reference to one or more registers.

The application schema describes relationships between those features

S100 application schemas shall be specified in UML Class diagrams from the UML model can be included in the product specification to explain or visualise those feature types which have complex relationships.

There are several ways a vector dataset can contain its feature, topology & spatial information: aggregate, complex, individually in line. The in-line approach hinders sharing geometry, and should therefore only be used in simple products. The application schema will document the approach to be used for a given product.

[12] ISO 19109:2005 Figure 4

Aggregation

Comment [Peter Par9]: Change examples to say “Feature Collection” rather than “Data Collection”?

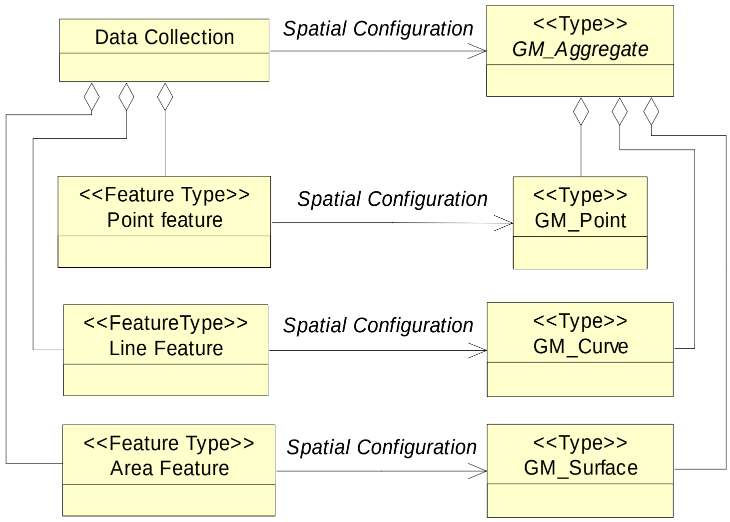


Figure 4 — Aggregation

e.g. Multi-primitives

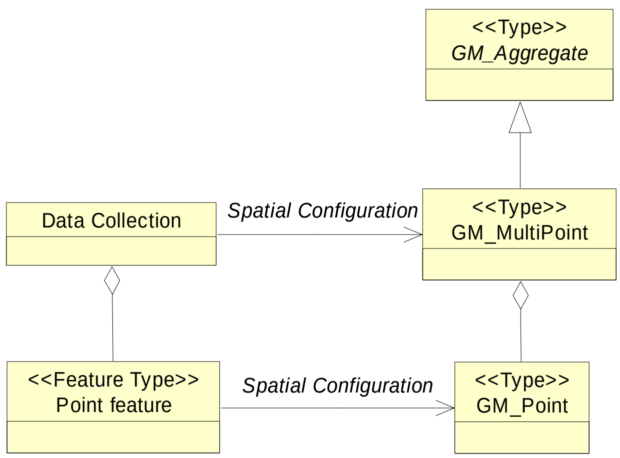


Figure 5 — Point features

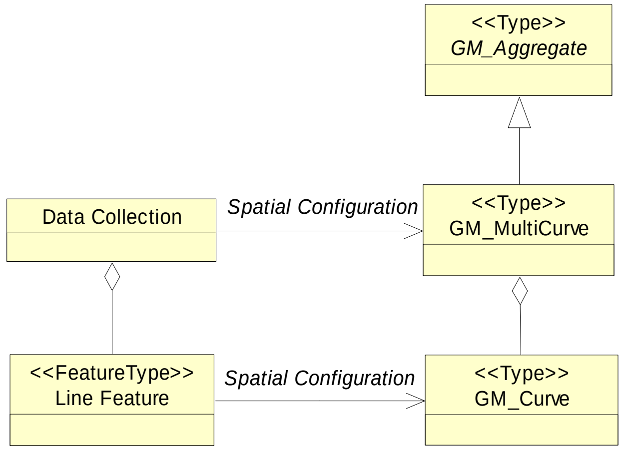


Figure 6 — Line features

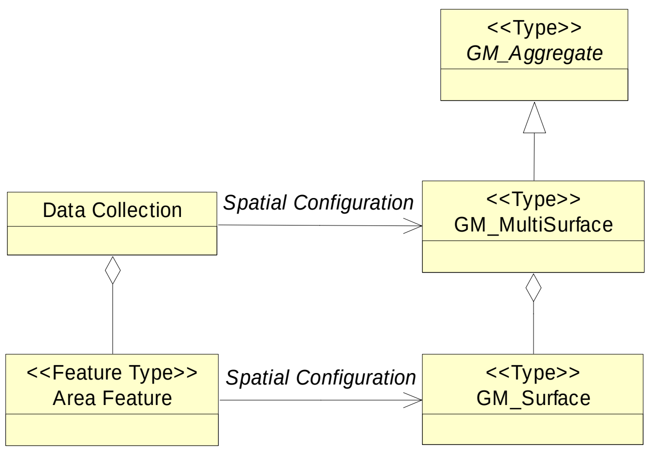
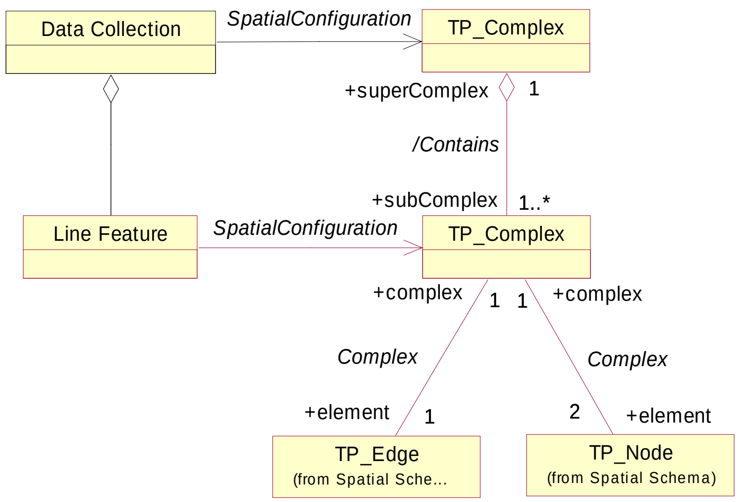


Figure 7 — Area features

(Topology) Complex



Comment [Peter Par10]: Replace with a "Geometry Complex" example? Depending on decision on topology…

Figure 8 — Line features

Where an S-100 product includes or consists of imagery and / or gridded data the model will be different. This approach should be chosen where the gridded nature of the data means a compatible content model is more important than the particular phenomenon (feature) represented. This may be because the application use case involves manipulating the data or portraying it, making use of grid or TIN handling algorithms. See the S-100 Imagery and Gridded Data (IGD) Component for more details.

If the grid simply carries values of a phenomenon (e.g. sounding, salinity, ocean current), the application schema is likely to follow this template, or one of the similar ones in the IGD component:

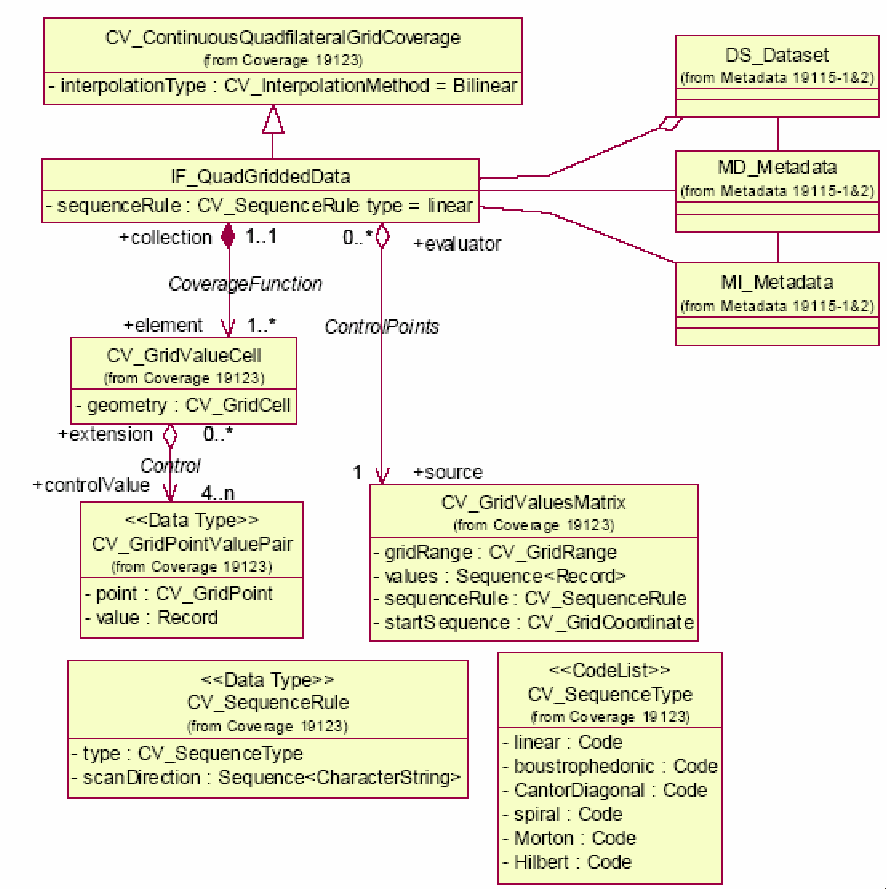


Figure 9 Gridded Data[13]

In this case, the coverage is itself the feature (the coverage carries the feature code, linking it to the feature catalogue). Alternatively, a feature could relate to the entire grid, using it as the value of a (spatial) attribute.

If the dataset contains vector features which link to particular point within the grid, e.g particular pixels within an image (cf GML and JPEG 2000). The IGD component includes this model:

[13] Copy of Figure 16 in the S-100 IGD Component

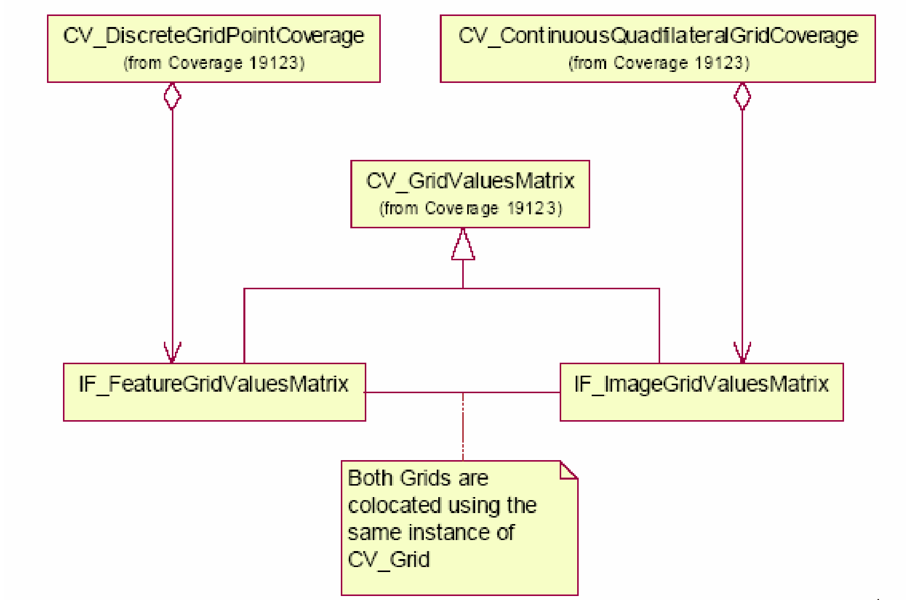


Figure 10 Assigning feature codes to pixels in an image[14]

6.3 *The General Feature Model (19109 Clause 7.3)*

6.3.1 *Introduction - the purpose of the GFM (19109 Clauses 7.3.1 and 7.3.2)*

The General Feature Model (Fig 2) is a meta model used to define feature types and their properties. It contains such concepts as feature attribute, feature association and feature operation. The General Feature Model also acts as a meta model for feature catalogues. The diagram included here has been cut down to only those components which are to be used in S-100; the clauses following discuss how to use them. For a discussion of why other components from the standard GFM are not included, see .

6.3.2 *The Main Structure of the GFM (19109 Clause 7.3.3)*

See Annex A for a discussion of the relationship between this S-100 model and the ISO 19109 General Feature Model.

[14] Copy of Figure F2 in S-100 Imagery and Gridded Data component

Comment [PP11]: To be replaced with an updated version (this dates from New Zealand)

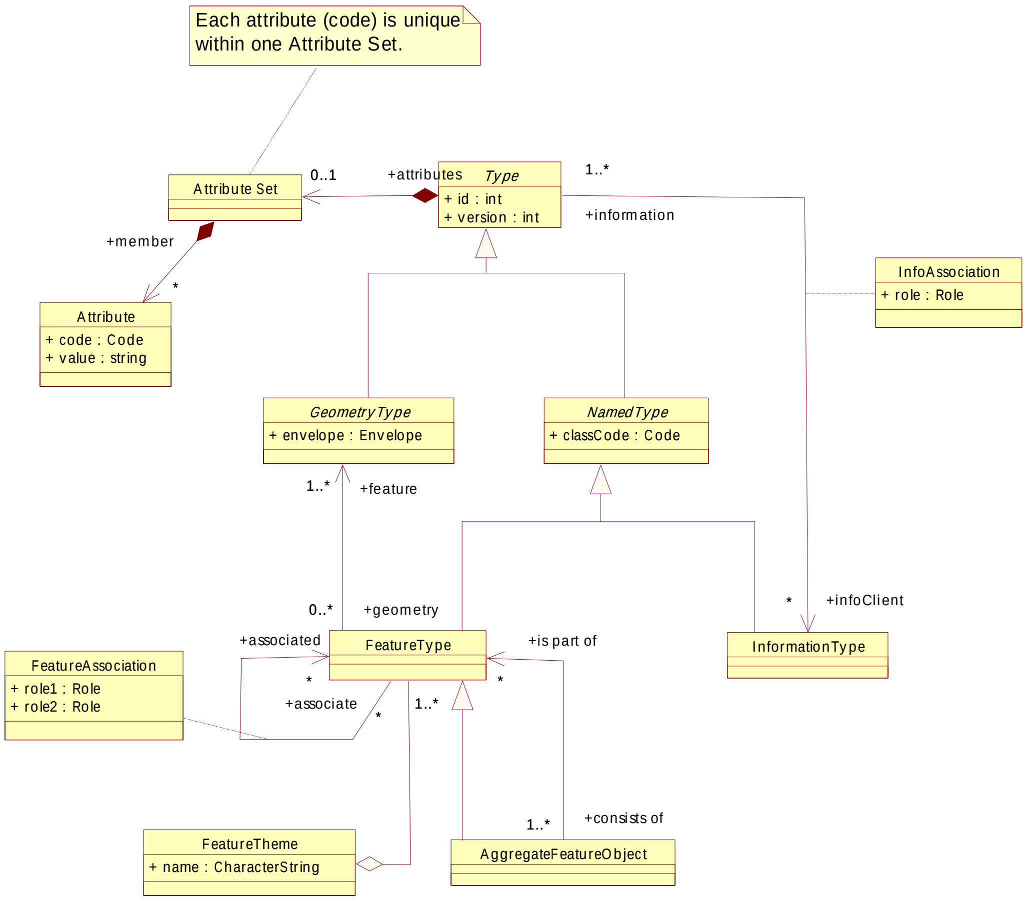


Figure 11 The General Feature Model

The biggest changes are to allow spatial objects to carry attributes, and the introduction of information objects (the InformationType) to handle certain complex feature properties.

Note: the presence here of NamedType.classCode and Attribute.code leads on to the 'weakly typed' approach, which is not common in the other ISO standards (e.g. particularly 19136)

6.3.3 *GF_FeatureType (19109 Clause 7.3.4)*

The actual feature types allowed in a given application will be catalogued, in accordance with ISO 19110 Methodology for Feature Cataloguing. The Feature Catalogue for a given application schema will reference one or more registers, e.g. the IHO Feature Data Dictionary.

S-100 Feature type names (GF_FeatureType.typeName) shall only use characters within ISO646.

6.3.4 *GF_PropertyType (19109 Clause 7.3.5)*

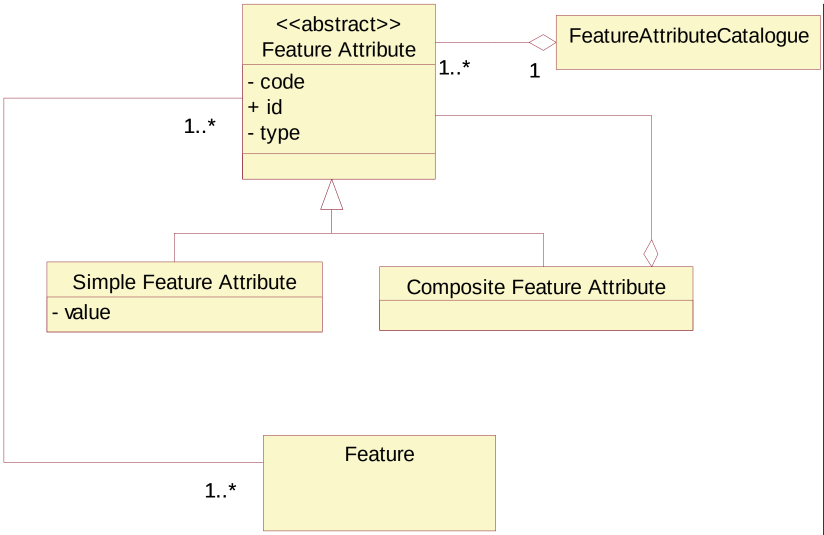
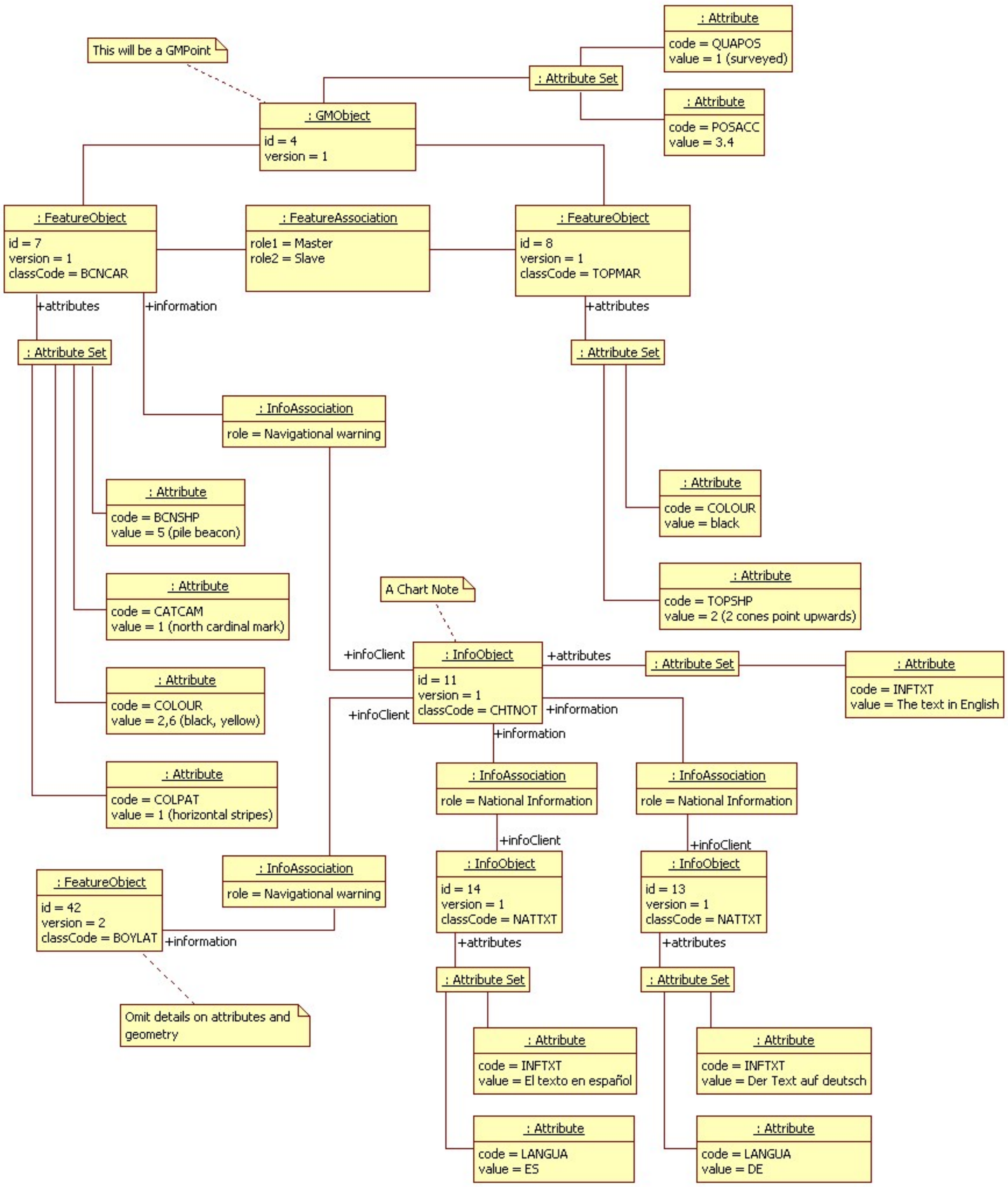


Figure 12 S-100 Properties

Comment [Peter Par12]: This should be replaced, but perhaps Holger's "Example" object diagram is a bit complex to stand alone?

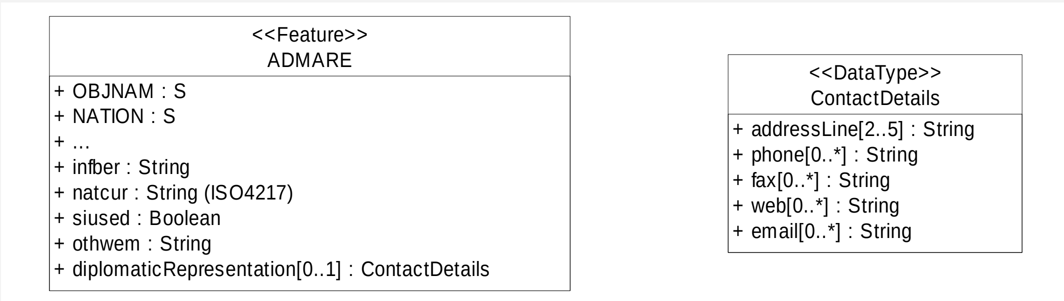


Comment [Peter Par13]: Need a revised export from StarUML, which would reflect the name changes.

S-100 property type names (GF_PropertyType.memberName) shall only use characters within ISO646.

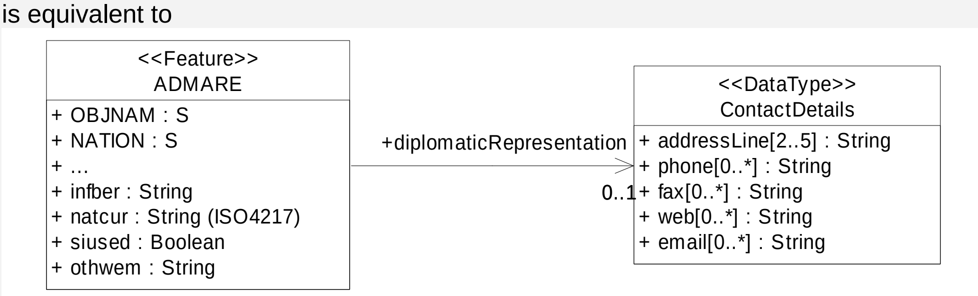
Feature property types are considered to be either associations or attributes. Attributes (e.g. GF_AttributeType.valueType) may be complex (composite) or simple (primitive). Complex types shall be used in preference to AttributeOfAttribute. Complex types shall be documented in the application schema, showing how they are built from the simple types.

Example: a product specification which needs to include addresses could reference an address model from an external register, or create one of its own - perhaps simply a collection of strings. This class can then be referenced as the data type of an attribute, and implemented as a class associated (by named role) with the relevant feature class.



Comment [Peter Par14]: Replace with ContactDetails structure for which I'm expecting a proposal from SNPWG

is equivalent to



In both cases, addressLine, phone, fax, web, email and ContactDetails would be registered in one or more data dictionaries (in addition to ADMARE and its other properties). The Feature Catalogue would specify which definitions to use, and that the ContactDetails must contain 2 to 5 addressLines (remember, this is a hypothetical example!), and that an ADMARE may have a "diplomaticRepresentation" of type (or, which associates to) ContactDetails. The semantic definition of "diplomaticRepresentation" would state that it is the diplomatic representation of the producing agency's country in that ADMARE.

Where a property:

- has a simple (primitive) type
- has a maximum multiplicity of 1
- need not have its value shared between (feature) objects

it can be modelled as a UML attribute. In GML this would be encoded as a GML property "by value", in a "non-GML" XML, it could be encoded as an XML attribute.

Comment [PMP15]: True, but perhaps distracting/confusing.

If any of these conditions is false – i.e. the property type is complex, the maximum multiplicity is greater than one, or the property value may need to be shared between more than one feature object (of the same or different feature classes), then the property could be modelled as a named association to an "information object" (UML class not stereotyped as features, i.e. not a realization of GF_FeatureType), or could be modelled as a data type. These would then be modelled as GML properties (XML child elements), with information objects probably by reference, and data types generally in line.

Comment [Peter Par16]: Discussion still open: are we allowing two models of attribute/property?

Simple properties could be modelled in the same way, to allow a consistent updating mechanism.

Comment [PMP17]: I think we're not doing this…

One of the reasons why some properties may have more than one value is to allow for multiple languages. The "information object" model copes well with this. In some cases, it may be appropriate to have a "simple" property in a default language (specified in the metadata), and one or more "national" (other) languages as a repeating property. In other cases, a symmetrical approach may be more appropriate, where (*all, many, some?*) human readable text properties are handled as information objects, with a language attribute. *Note: the draft metadata component recommends dividing the (meta)data set up into separate sections, each of which has a consistent language.*

Comment [Peter Par18]: Again, do we specify one approach at S-100?

In the example above (Figure 9), the dataset has a default language of English (defined by the product specification and/or by the dataset metadata), at least for its "INFTXT" attributes; other languages are carried using Information Objects (NATTXT) which carry a language attribute as well as the INFTXT. This example uses a weakly typed encoding, but the same language handling could be implemented in a strongly typed encoding. *Alternative (not yet ruled out?) is to have all languages equal*

Attributes can have a multiplicity greater than one, which – at a logical level – implies the attribute is repeated within the set. Some encodings (e.g. GML) have no problem with this; others (8211?) do, and require multiple attribute values to be handled in a product specific way, e.g. as coded strings. All cases where a feature instance may have multiple occurrences of the same attribute make it difficult to update individual attribute values. If this is required, the Product Specification shall clearly state how it is to be achieved (e.g. ordered values).

Attributes (from the feature catalogue) shall be identifiable in any encoding.

6.3.5 *GF_AttributeType (19109 Clause 7.3.6)*

This refers to attributes representing the properties of features ("Feature Attributes"), not to any attributes required to manage a specific encoding.

Allowable primitive types (GF_AttributeType.valueType) are a subset of those defined in ISO 19103, and are listed at Annex B. If an Application Schema developer needs to use an additional attribute type but remain part of the "S-100 family", this would be a change to the S-100 standard. IHO S-100 has chosen not to manage data types in a register, because introducing new data types to the standard requires greater control than introducing new feature or attribute types – it is hard to create software which could dynamically handle unexpected data types!

The actual attributes allowed in a given application will be catalogued, with reference to one or more registers.

Cardinality is a combination of the concepts “optionality” and “multiplicity”, that is, if the value of the cardinality includes (starts with) a zero, the attribute is optional. Attribute cardinality (especially maximum multiplicity) may be specified in the data dictionary. If not, or if the application schema wants to further constrain it, the cardinality shall be specified in the feature catalogue.

Some value constraints appear in the Feature Data Dictionary (e.g. bearings are angles<=360). Other simple constraints shall be defined in the Feature Catalogue (e.g. minimum and maximum values, limitations on enumerated values).

Those constraints which are too complex to express in the registers shall be documented in the Product Specification (data capture section).

(ENC: Need to revisit all S57 types e.g. formatted text types & (preferably) re-model them as types. And then re-visit the “use of the object catalogue”)

6.3.6 *Associations (19109 Clause 7.3.7 GF_AssociationRole and 7.3.9 GF_AssociationType)*

This clause is about named associations between feature classes (i.e. it doesn’t include the associations between feature classes and data types - information object classes - described above). Associations should be of real world navigational significance. *Could consider associations to support portrayal rules (topology).* Try to avoid completely generic associations (without nameable roles).

Each application schema shall model all its associations. The association roles shall be named, catalogued and registered (in the FDD), with (preferably) a recommended complement role. Actual binding of the roles to one another, creating the association, and to the classes, takes place in the feature catalogue - it is product specific.

> Example: a Data Dictionary could include definitions for roles “spans” and “spannedBy”. The same (or another) dictionary could define bridge, road, river, canal, railway. The Feature Catalogue would define that a bridge may span zero or more roads, rivers, canals or railways, and that each of these feature classes may be spannedBy zero or more bridges.

The Product Specification / Application Schema will say how it will be realised in an encoding. Generally they would be implemented by implementing the roles as properties of the classes, rather than as association classes. Association classes should only be used for many-to-many associations.

Again, some binding/constraints may be too complex for the Catalogue, and may only be described in the Product Specification.

> Example: a product specification may want to state that a bridge may only have a vertical clearance property if it also has a “spans” association with some kind of navigable water.

Only use binary associations (i.e not more than two roles!); this means that each instance of the association can only associate two feature objects. They may be of the same or different feature types (classes). GF_AssociationType.includes retains a cardinality of 1..*, because the same association may be used between objects of many different classes, it is just that each instance of the association can only be between two objects.

6.3.7 *GF_InheritanceRelation (19109 Clause 7.3.10)*

Note that in pure theoretical UML and in section 7.5 of ISO 19109, inheritance is seen as one kind of relationship between classes. See 1.24.1 for further discussion.

6.4 *Attributes of Feature types (19109 Clause 7.4)*

The actual attributes allowed in a given application will be catalogued, with reference to one or more registers.

6.4.1 *Spatial Attribute Type (19109 Clause 7.4.3)*

In the TC211 standards the geometric properties of a feature are implemented as associations to spatial (geometry) classes; this allows spatial objects to be shared, leading to more efficient datasets. S-100 component xxxx describes the allowable spatial objects, and how they relate to one another. This document will include examples of how given application schemas should use the spatial objects.

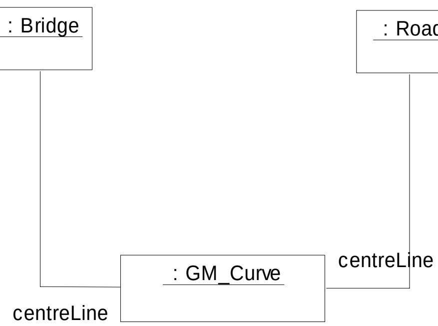


Figure 13 — Example of features sharing geometry

A feature object can have several spatial (geometric) properties, describing its position and/or extent. For example, a building may have a 'footprint' (which would be an area) and an 'entrance' (which is likely to be a point); or an archipelagic sea-lane may have a centre line and also (advisory) limits (two further lines, or an area). For this reason, the feature catalogue shall document these named associations to spatial objects (named geometric properties). Of course, it is always possible to model these as several separate but associated objects, each with a single spatial property. *Should S-100 mandate one approach?*

S-100 includes a profile of ISO 19107, via the Harmonised Spatial Profile (HSP) drafted jointly by the IHO and DGIWG. This profile consists of Simple geometry (spaghetti), Chain Node, Edge Node Graph, Planar Chain Node and Full Planar Graph. The HSP introduces "implementation classes", which realise properties of both geometric and topological classes from ISO 19107. *S-100 could extend these implementation classes to include attributes for spatial quality - but this may impact on interoperability with general TC211/GML software, which at best would miss the quality attribution.*

Comment [Peter Par19]: Spatial profile (or CRS profile?) should include a note about the problems of line interpolations in different projections. If not, make sure it's in here.

The models in this document are high level; more detailed information of individual elements can be found in Part xxxx the IHO Spatial Profile.

Geometric & topological primitives & associations are not kept in a register; any changes would need to be handled as changes to the standard. This is because changes to the geometry model would have a much larger impact on the producing & consuming software than changes to other attribute definitions.

An S-100 Feature Catalogue shall specify

- which spatial object classes are allowed
- which feature classes may associate with each particular spatial type. This may be inherited from the register providing the definition of the feature class, or the application schema may restrict the classes further.
- may make further constraints, e.g. constrain curves to having only one segment, or constrain line types/interpolations.

> For example, a buoy in the data dictionary (register) may be able to associate with either an area or a point. An individual product specification (knowing the maximum scale it will be used for) may constrain buoys to points.

Comment [Peter Par20]: do we have a separate register for coverage (soft) feature types? Probably an enumerant…Depth area with a geometry of type TIN, vs TIN of depth measurements.

An S-100 Application Schema:

- can only use geometry models from the IHO profile
- will state which models may be used
- shall document any use made of orientation (see below)

6.4.1.1 *Shared Lines and Orientation*

When using curves to delineate an area, 19107 defaults to "left is on the inside", i.e. if the orientation is positive, the curve surrounds the area anti-clockwise. Traditionally, GIS and navigational systems have taken the opposite approach, with the boundary going clockwise. In either case, 19107 allows two "Rings" to share part of their boundary (e.g. "LineString") via OrientableCurves with opposite orientation:

Comment [Peter Par21]: We have three proposed methods - choose at S-100 level?

Comment [Peter Par22]: Sections deleted here, which will be in the spatial & IGD components.

6.4.2 *Temporal Attribute Type (19109 Clause 7.4.4)*

See for allowable ISO 19103 date and time types. All dates will use the Gregorian calendar. Two temporal references systems are allowed: UTC (with time zone) and "local"; an application schema may restrict this further. The feature catalogue shall specify the time type for each attribute.

At present, we assume that none of the "advanced" types discussed in ISO 19108 will be required.

6.4.3 *Quality Attribute Type (19109 Clause 7.4.5)*

This is for attaching quality statements to individual feature classes (including feature collections). See S-100 Component Metadata - Part 2 for details of the IHO feature metadata profile.

In S-57, the IHO believes it has a good model for positional quality - extending the spatial object classes to allow quality attributes there, rather than on the associated feature object.

Comment [Peter Par23]: Can this allow "vague geometries", which may be required for SD passages?

Extend spatial objects to have quality (presumably at some suitably abstract level); encode in such a way that it can be read by a non-IHO GML reader!

Comment [Peter Par24]: Presuma bly addressed in the spatial component.

6.4.4 *Metadata Attribute Type (19109 Clause 7.4.7)*

IHO is producing a discovery metadata register, some parts of which will be mandatory. *The rules about which are mandatory will appear in this document.* The register will be extended to cover other parts of 19115, as required.

Each application schema (feature catalogue) will reference elements from this register.

Comment [Peter Par25]: Does the catalogue now become a multi-part catalogue, with feature, meta,…?

Can use FeatureCollection->Meta or MetaFeature objects (geometric containment) to reference metadata at the dataset or feature collection level.

Can use individual links from feature->meta.

Metadata may also be used at the feature attribute levels. This section describes the various elements for describing data at the dataset level. Dataset metadata shall be described in the Product Specification, see for further information.

S-101 TO DO: sort out the S-57 features & attributes which will actually be metadata items.

*Include an example, probably of one of the current M_*

Bring text in here from the scope of Metadata parts 1 & 2.

Should we move to a range of M_QUALs (or a soft typed one), which indicate which classes it applies to?

6.4.5 *Thematic Attribute Type (19109 Clause 7.4.8)*

In ISO 19109 this means the attributes which are relevant to the theme of the data, as opposed to the position, quality and metadata attributes. That is, this covers all the attributes used to model the relevant properties of the real world object. The Feature Catalogue will bind attributes to classes. See 6.3.4 GF_PropertyType (19109 Clause 7.3.5) for discussion of attributes.

Application schemas may want to group the features into themes; this could be done by means of FeatureCollections, aggregation associations, inheritance or thematic (group) attributes.

S-100 Application Schemas shall model these themes / groups (“layers”) by using an attribute on each feature class which could be classified. The Product Specification shall state which feature classes should be in each group.

For example, register an attribute “myGroup”, which is an enumerated type, and register the enumerated values, starting with “Group 1: skin of the earth”. This application schema indicates that the attribute is available to all Feature Classes, whereas the “Colour” property is specifically used with the Light class.

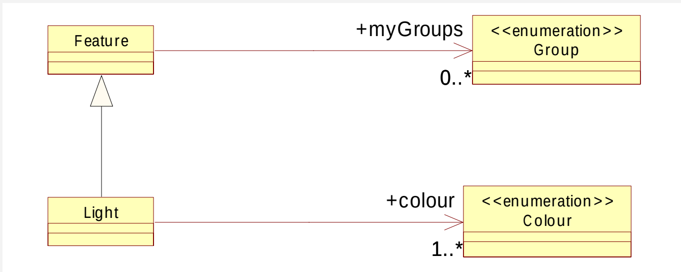


Figure 14 - Modelling Themes using enumerated attribute

To do: include another example & diagram.

This is then the same as any other enumerated attribute of a feature class. The product specification will include the rules about what each group should contain – these should be included / referenced in the definition of the enumerated value in the Feature Data Dictionary.

6.5 *Relationships between feature types (19109 Clause 7.5)*

6.5.1 *GF_InheritanceRelation (19109 Clause 7.5.2)*

This section illustrates the use of inheritance. It uses, as an example, a series of diagrams based on the components of a system of navigation marks.

The abstract super class FT_NavigationalMarkComponents contains all the generic attribution required by the sub-classes in the complete inheritance tree.

The two abstract classes FT_Buoy and FT_Beacon inherit from FT_NavigationalMarkStructure and add the generic attribution required by their sub-classes. The remaining concrete sub-classes contain the specific attribution required for each individual feature type.

For a dataset model, this is logically equivalent to copying all eight attributes to each of the eleven concrete classes - in a dataset (instance), all the attribution is carried at the lowest level – objects instantiated from concrete classes. Abstract classes can not exist in data instances. Inheritance in the application schema, feature catalogue or data dictionary mainly provides hints to system manufacturers that groups of feature classes have certain behaviour in common; it can also reduce the size of the feature catalogue.

Comment [Peter Par26]: So, how relevant is this to S-100?

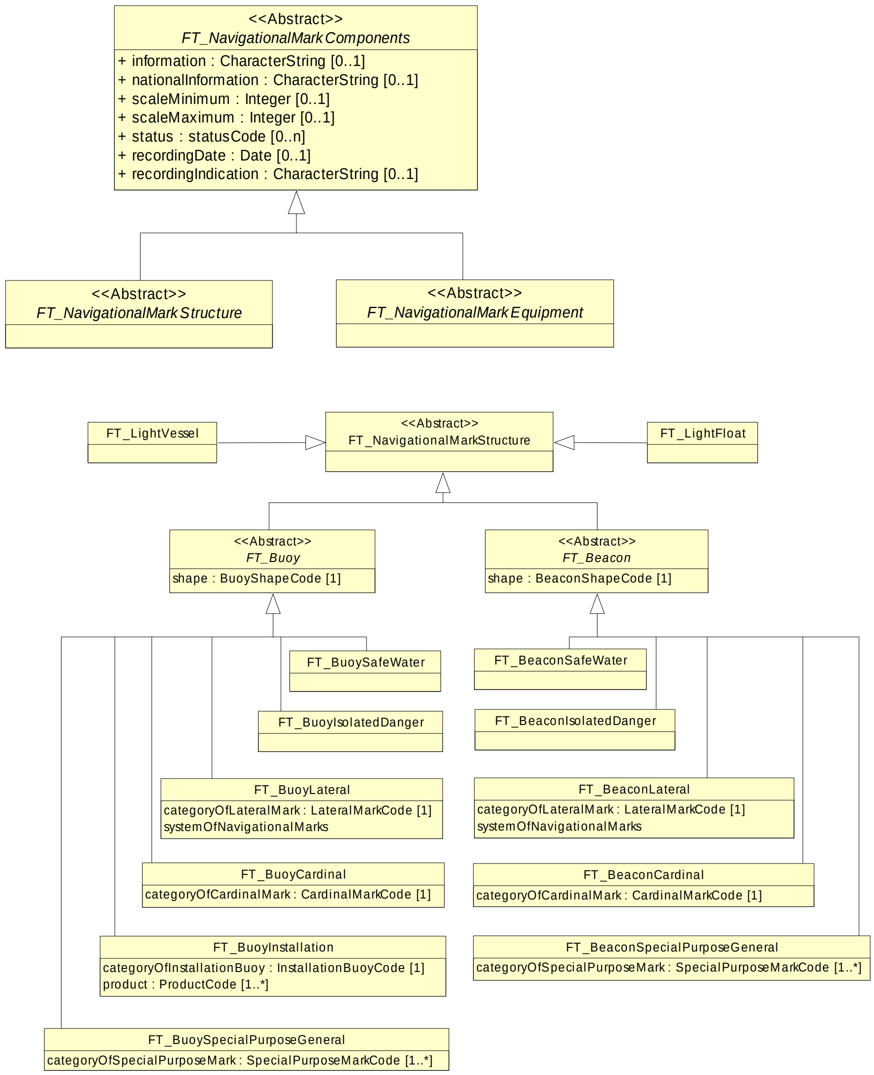


Figure 15 Example of inheritance

6.5.2 *GF_AssociationType (19109 Clause 7.5.3)*

Other than inheritance, we consider three main types of association between feature objects: aggregation, spatial association, temporal association.

6.5.2.1 *GF_AggregationType*

This clause is about aggregations of features, not aggregations of spatial objects; it represents "whole-part" relationships. Figure 16 Example of composition and Figure 17 Example of weak aggregationcomposition illustrates two kinds of aggregation.

In the first example a navigational mark consists of one "structure" feature (e.g. a buoy sub-typed from FT_NavigationalMarkStructure or a pile from FT_Hydrography) and one or more "equipment" features (e.g. a light and a topmark from FT_NavigationalMarkEquipment). Whilst generic aggregations could be used be used, named aggregations ("comprised of") provide more functionality particularly for querying operations. In this example, the aggregation FT_NavigationMark is a concrete (as opposed to abstract) class. This example may be modelled using *composition* (strong aggregation), where if the FT_NavigationalMark is deleted, the two components are also deleted. Alternatively, an application schema could decide to have just one feature – the NavigationalMark, with the other parts modelled as associated properties ("information objects").

The feature classes FT_Hydrography and FT_LandTopography are included for completeness and will not be described further, other than that they include potential navigational structures i.e. piles, land marks etc. These would not use strong aggregation – the bridge support remains even if the warning light is removed.

Comment [Peter Par27]: The diagram is incorrect on this point…

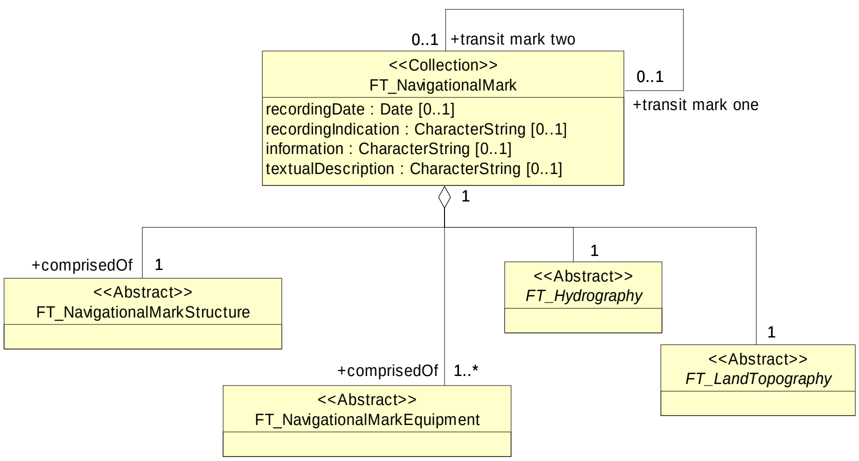


Figure 16 Example of composition

Comment [Peter Par28]: To be fixed to match once we have an actual model. In particular, this mixes composition and weak aggregation!

Routing measures example

A Traffic Separation Schema may be an example of "weak" aggregation - the Navigational Marks may not disappear if the TSS does, and vice versa.

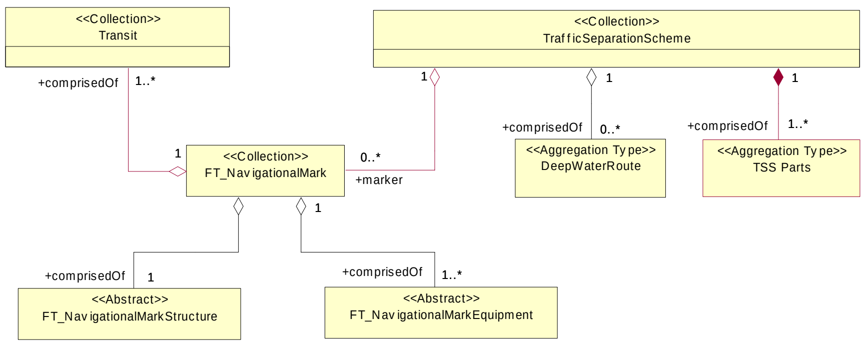


Figure 17 Example of weak aggregation

Encoding: Feature Collections

ISO 19136 encodes these aggregations using Feature Collections, and generally reserves the term "aggregate" (and related terms) for aggregations of spatial objects (the exceptions are: other composite property values, the AggregationAttributeGroup - which is shared by Feature Collections and Aggregates, and the AggregationType of a collection). Feature Collections are a specialisation of Feature, i.e. the collection itself is the representation of a real world object; the difference is that one or more of its properties are themselves features. Not all the members of a feature collection need be of the same class.

Where this is done, it is normal to gather the associated spatial objects into a spatial complex as well.

Comment [Peter Par29]: Should this be an S-100 rule - where feature objects are aggregated, the associated spatial objects shall also be aggregated / put in a complex?

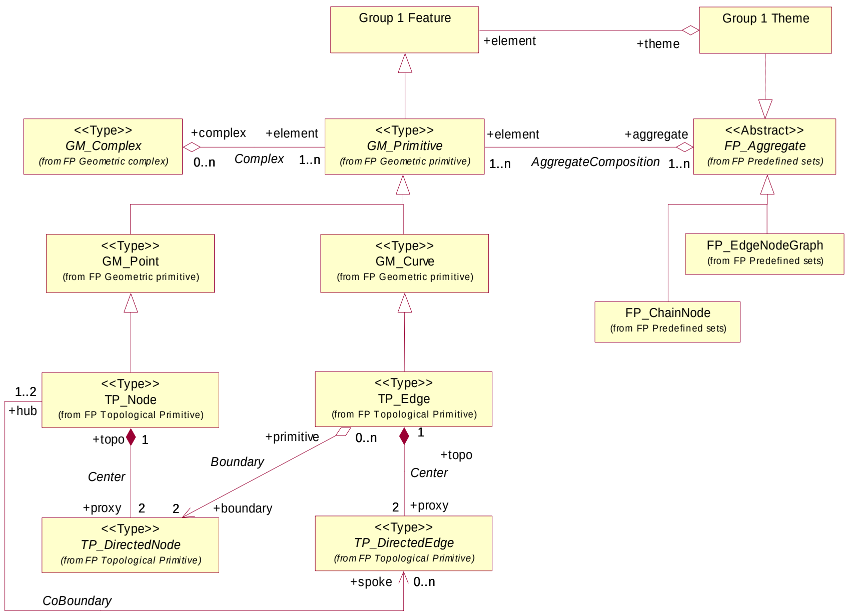


Figure 18 — Relationship between Feature Themes and Geometric Complexes

Comment [Peter Par30]: Fix: confusing to have "Group 1 Feature" as the supertype for the GM_...

However, Application Schema developers should avoid cross linking collections, i.e. situations where feature objects may belong to more than one collection. This would make the datasets difficult to maintain and update. Where this might happen, the Application Schema shall model the themes / groups using an enumeration attribute on each feature class, as described at 1.23.5.

Comment [Peter Par31]: Or are we preferring that model for all situations?

All Feature Collections / aggregates shall be in the Feature Catalogue.

6.5.2.2 GF_SpatialAssociationType

This section in ISO 19109 refers to explicit spatial ("north of", "above") and topological ("intersects", "touching") associations between features. Spatial associations may also be described implicitly using relationships between the associated spatial and topological objects.

Where an application requires explicit spatial associations between features, these shall be catalogued. Where features are associated by sharing geometry (or topology), this shall be clear in the application schema.

Annex A (normative)
Notes for a Profile of ISO 19109

This annex will document restrictions and S-100 specific usages clause by clause for ISO 19109

6.2

S-100 feature type definitions *shall* be documented in a feature catalogue.

7.2

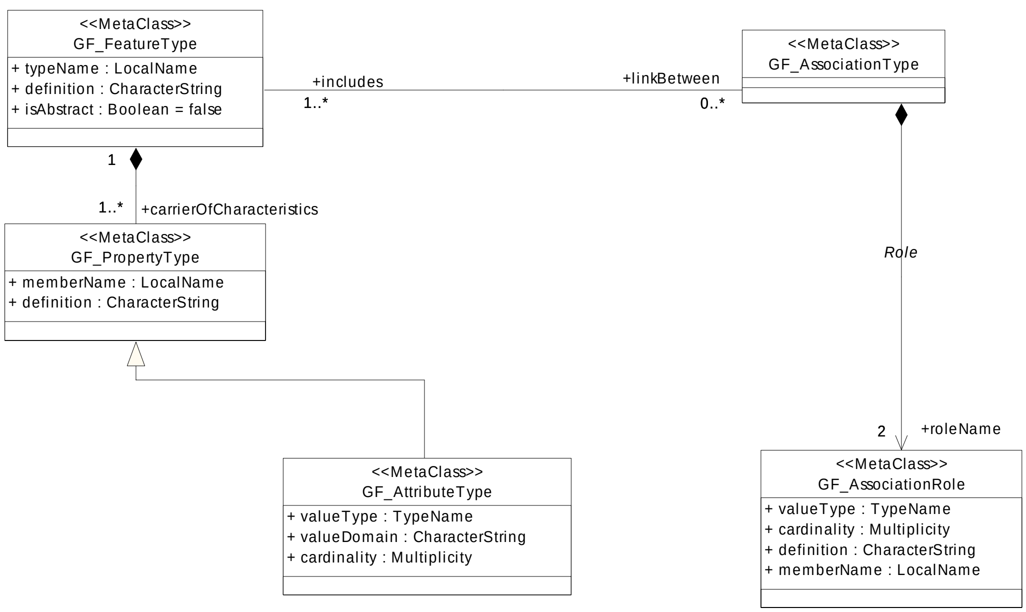
S-100 application schemas *shall* use feature type definitions from one or more registers.

S-100 application schemas *shall* be specified in UML.

7.3.2

S-100 feature type definitions *shall* be documented in a feature catalogue.

GFM changes (w.r.t Fig 5 in 19109):



[15]

Figure 19 The (Reduced) General Feature Model

---

15 Based on the diagram in the 19109 package of the All ISO Model, available from www.tc211.org

Changed multiplicity of GF_FeatureType->carrierOfCharacteristics to be 1 at the GF_FeatureType end; I believe the 0 is only there for the GFM to be used in data dictionaries (where a GF_PropertyType can live independent of any GF_FeatureType).

We restrict ourselves to binary associations and require both roles to be named, so the cardinality of GF_AssociationType.roleName is 2. The GFM allows two routes to associate GF_AssociationRole with GF_FeatureType; we drop the inherited “carrierOfCharacteristics” association, and retain the association via GF_AssociationType.

For S-100 application schemas, Attribute of Attribute will be implemented using complex attribute types. Attributes with complex types are equivalent to associations to (non-feature) classes - instances of which are sometimes known as “information objects”.

Drop operations & constraints

Don’t worry that GF_AssociationType is a specialization of GF_FeatureType

Drop GF_InheritanceRelation; inheritance is a useful concept for software development, and could be discussed in the data dictionary (as advice to software developers). However, inheritance can’t be stored in a data set, so it is not a focus for a product specification. (The cardinality of GF_InheritanceRelation permits the use of multiple inheritance e.g. a subtype can belong to more than one super type. Where possible, it is advisable to restrict the multiplicity to 0..1 due to the potential complexity of the ensuing model and implementation.)

GF_AssociationType and GF_AssociationRole are for associations between feature classes.

As the S-100 classes are no longer identical to the 19109 classes, we removed the GF_ prefix. We then introduced GeometryType, to allow geometry objects to carry attributes (in particular, quality), and finally introduced InformationType to handle structure properties.

7.3.3

For S-100 application schemas, Attribute of Attribute will be implemented using complex attribute types, (except in the backward-compatible 8211 model within the S-101 ENC Product Specification).

GF_AssociationType need never be used, although it is logically / implicitly present; S-100 application schemas shall catalogue and use GF_AssociationRole. GF_AssociationType.roleName shall have a cardinality of 2.

7.3.4

S-100 feature type names (GF_FeatureType.typeName) shall only use characters within ISO646.

7.3.5

S-100 property type names (GF_PropertyType.memberName) shall only use characters within ISO646.

7.3.6

Attribute cardinality (GF_AttributeType.cardinality) may be specified in the data dictionary. If not, or if the application schema wants to constrain it, the cardinality shall be specified in the feature catalogue.

7.3.8 GF_Operation and 7.6 Behaviour of Feature types

Arguably, GF_Operation could carry the interface to a web service, for example a tide-gauge reporting service, or a real-time tidal prediction service. However, S-100 is about static data products; the data items will not modify themselves as a result of any operation, so all potential operations can be modelled as attributes or associations. S-100 therefore excludes operations.

7.3.9

The GFM sense of GF_AssociationType being a subclass of GF_FeatureType may be of little use in S-100 – except to indicate that associations exist in the Feature Data Dictionary.

7.3.11 GF_Constraint and 7.7 Constraints

Value constraints will be documented in the data dictionary, catalogue, or product specification. We do not envisage carrying constraint information in S-100 datasets, and have therefore not included GF_Constraint in the S-100 feature model.

7.4 attributeOfAttribute (19109 Clause 7.4.2)

This describes an association between an attribute and one or more others, which characterizes it. This was common in S-57 (VALSOU, QUASOU & TECSOU; NATSUR, NATQUA), but not made explicit. However, there are always other ways to model this, using associations to classes which carry all the information. This approach - complex data types - will be used for S-100 application schemas, except S-101 where required for backwards compatibility with ISO 8211 encoding.

Comment [Peter Par32]: Julia: Use an example, but do not refer to S-57

Comment [Peter Par33]: How to handle these exceptions, where we only want it in the Framework for this one use, but don't want to distract future readers with the past…

7.4.6

Location Attribute Type shall not be used. Could be included in a future version.

7.4.8

NOTE: Information about S-100 thematic attributes *shall* be found in *the* feature catalogue.

6.5.2.3 7.5.3 GF_TemporalAssociationType

This could be used to model succession. At present, we haven't seen reason to include it.

8.2.1

The conceptual schema language for S-100 application schemas shall be UML.

8.8.2

An S-100 application schema *shall* be completely constructed from the definitions provided by the feature catalogue.

Annex B (normative) Primitive Data Types

This annex documents those primitive data types which are allowed in S-100 data sets. The detailed definitions are in ISO 19103, and ISO 19136 provides mappings to xsd or gml data types.

Integer

Real

Vector

Date, Time, DateTime, from 19103. These should be encoded as per ISO 8601, not using a more structured encoding such as might be caused by applying ISO 19118 or other rules to the UML class diagrams in 19103.

CharacterString with CharacterSetCode defaulted in metadata, S-100 **default** to 10646-1 encoded in UTF-8 – backward map to 646 / ASCII for backward-compatible lexical level 1 output NOTE for class/property names: only use ISO646; language could be defaulted as well?;

Boolean; sign only where required for topology…; multiplicities may be in the DD or FC.

Specialisations of Measure: Length, Angle. Could we add “Mass” here?

Probably not Decimal

Probably not unlimitedInteger (except perhaps in the FDD, where they want to use 0 for infinity?)

FormattedText - the intention is to hold strings which are formatted in xHTML (or other XML?); the basic type in ISO 19103 or 19136 would be CharacterString, and the formatting schema is a description of the domain.

Annex C Dataset Metadata

Metadata may be used at dataset, feature and feature attribute levels. This section describes the various elements for describing data at the dataset level. The Metadata Entity Set diagram (Figure 13) is included for information and gives an overview of the various metadata elements. A full description of metadata can be found in ?????? the metadata profile.

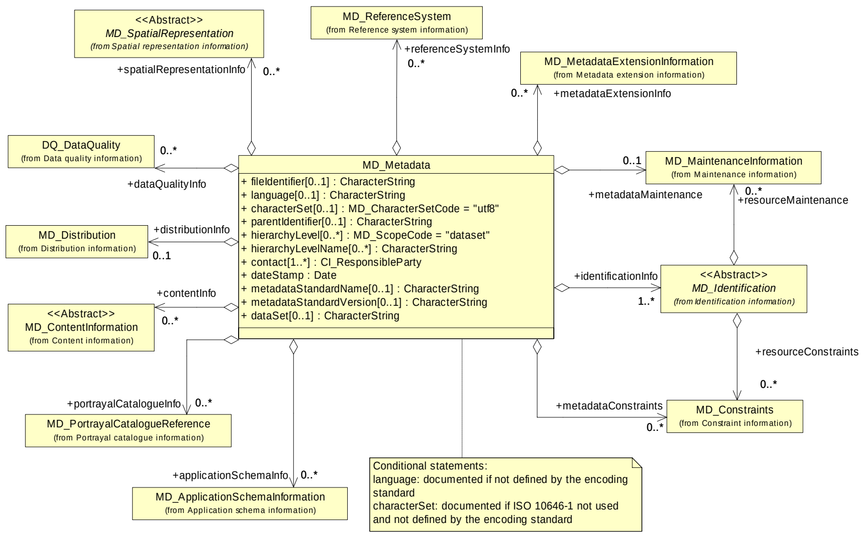


Figure 20 — Metadata Entity Set

A.1 Dataset identification – name, date etc

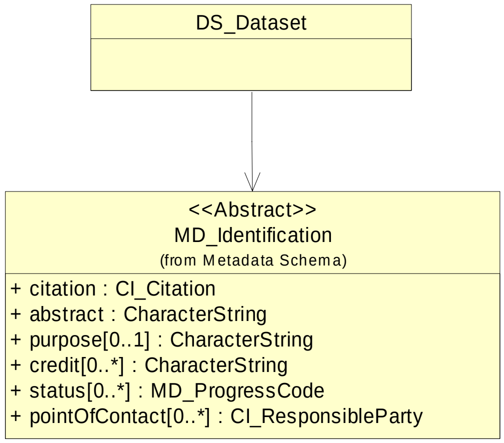


Figure 21 — Dataset Identification

A.2 Data identification

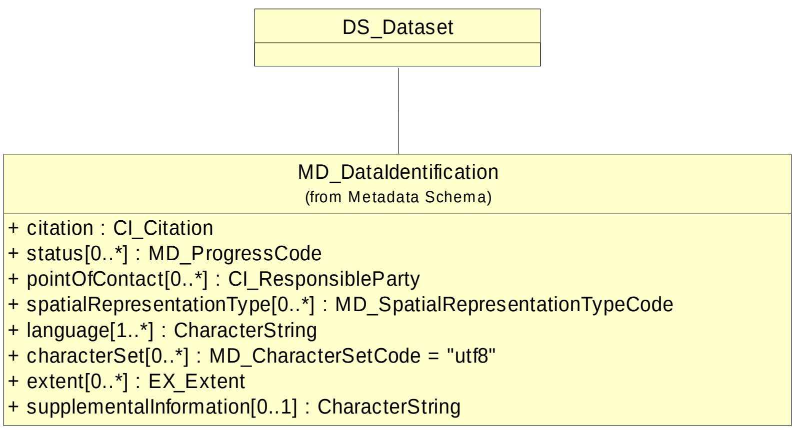


Figure 22 — Defining the extent and description of a dataset

A.3 Reference Systems

The application schema will usually specify the required or allowed reference systems, by reference to a CRS dictionary.

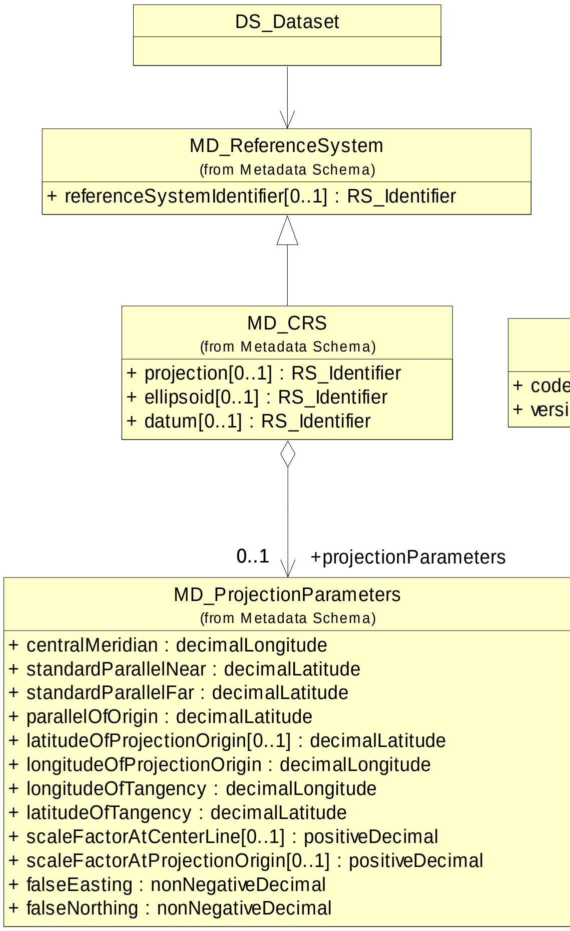


Figure 23 — Coordinate reference system metadata

A.4 Dataset quality

The IHO could create a register of quality evaluation methods, and the application schemas would then reference this.

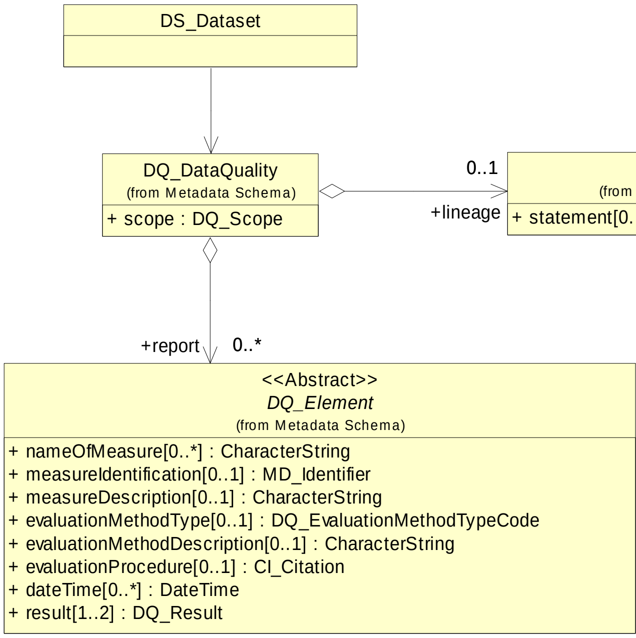


Figure 24 — Dataset Quality

Annex D Spatial Configurations / S-57 Geometries

Comment [Peter Par34]: Deleted, contents replaced by the spatial component - to be provided by Barrie?