Table of Contents

[FEATURE MODEL: 1](#_Toc362858612)

[General Feature Model 2](#_Toc362858613)

[Preliminaries 2](#_Toc362858614)

[Meta-model 2](#_Toc362858615)

[Formalizing a feature-type 2](#_Toc362858616)

[Encoding and transfer 4](#_Toc362858617)

[Information Communities and feature catalogs 4](#_Toc362858618)

[Mapping to other modeling formalisms 4](#_Toc362858619)

[Features and conventional GIS 4](#_Toc362858620)

[Tables, Objects and XML 4](#_Toc362858621)

[Example 5](#_Toc362858622)

[Geo Primitives, Composition and Aggregates (ISO 19107) 7](#_Toc362858623)

[Core Definition Model: 9](#_Toc362858624)

[Electoral Units 11](#_Toc362858625)

[Geofabric data products - an analysis of processes, descriptions and models 11](#_Toc362858626)

# FEATURE MODEL:

LINK: <https://www.seegrid.csiro.au/wiki/AppSchemas/FeatureModel>

The 'Feature' is the fundamental unit of geospatial information, so the Feature Model is the fundamental meta-model used for developing an Application Schema.

Features are typed objects with identity. This is often referred to as "vector" data in traditional GIS.

Feature types are defined by a characteristic set of properties (i.e. their attributes, associations, operations). A feature type is usually specific to an application domain, and will be part of a Feature Type Catalogue (FTC) that describes a key part of the language of a domain. Features often correspond with objects that are recognisable in the real world, such as road, mine, truck, storm. However, spatial properties are not mandatory, so a feature type could be defined for any item of interest within a domain. This potentially allows data access for both spatial and non-spatial information to be unified through a common interface.

The *General Feature Model* is formally defined in [ISO 19101](http://www.isotc211.org/Outreach/Overview/Factsheet_19101.pdf) and [ISO 19109](http://www.isotc211.org/Outreach/Overview/Factsheet_19109.pdf). For a more detailed discussion, see [FeatureModel](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/FeatureModel).

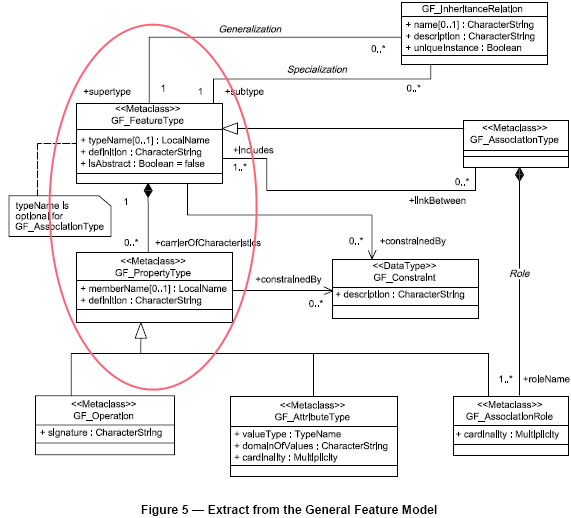
General Feature Model

Preliminaries

In accordance with ISO 19103, UML is used as the Conceptual Schema Language.

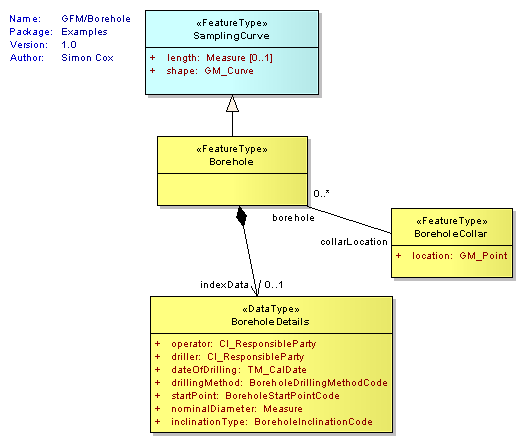
Meta-model

The General Feature Model is a "meta-model" for the definition of a **feature type** (i.e. the schema for feature instances) .



### Formalizing a feature-type

The definition of a feature-type may be formalized in a UML model. A UML profile is provided that constrains the variation that is allowed by generic UML, and adds some customization in the form of stereotypes and tagged values (i.e. the standard UML extensibility points) - see [SchemaFormalization#ISO\_TC\_211\_Profile\_of\_UML](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/SchemaFormalization#ISO_TC_211_Profile_of_UML).

This model for "Borehole" feature-type shows many of the standard patterns:   


This class diagram says that Borehole is a feature-type that specializes SamplingCurve. Two attributes are inherited: length which is of type Measure (a scaled number), and shape which is of type GM\_Curve (a bent line). Two navigable associations are added: indexData holds the BoreholeDetails in a data structure; collarLocation provides a link to another feature-type,BoreholeCollar.

Note the following:

* properties are shown as class attributes and associations
  + no operations are shown; all properties are realized as static values
* the set of properties is characteristic of the feature type.
* in the case of class attributes
  + the attribute name instantiates the memberName attribute of GF\_PropertyType
  + the attribute type instantiates the valueType attribute of GF\_AttributeType
* in the case of class associations
  + every navigable association-end carries a rolename
  + the association rolename instantiates the memberName attribute of GF\_PropertyType
* the type of a property may be structured
  + if the value of a property varies in the scope of a feature-instance, then it may be expressed as a function or coverage - see [OGCInformationModels#Coverages](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/OGCInformationModels" \l "Coverages).

For more detail on the mechanics of modeling feature-types, and a UML template pre-loaded with types from the ISO 19100 specifications, see [HollowWorld](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/HollowWorld).

For an example of a large domain-specific feature model, see [GeoSciML](http://geosciml.org/" \t "_top).

### Encoding and transfer

A feature model described using the patterns outlined above may be converted to a GML-conformant XML Schema. This defines a document-model for serializing instances of geospatial data, suitable for transfer. The UML-XML encoding rule results in an encoding within which the model structure is shown explicitly - see [GmlImplementation](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/GmlImplementation).

The OGC Web Feature Service defines an interface for requesting feature data, in which the query is expressed in terms of the GML representations, and services are required to provide a GML-encoded response, alongside other formats if desired - see [OGCServiceInterfaces#Web\_Feature\_Service\_OGC\_WFS](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/OGCServiceInterfaces" \l "Web_Feature_Service_OGC_WFS).

## Information Communities and feature catalogs

Communication within a specific information community is characterised by the information model and the **catalogue of feature types** that is agreed by its members (see ISO 19110, 19126).

## Mapping to other modeling formalisms

### Features and conventional GIS

The General Feature Model takes an object-oriented view of the world. To describe an object its "type", or the "class" to which it belongs, must be determined. This fixes what properties are associated with it, one or more of which may be geometric or spatial.

This approach contrasts subtly but importantly with the conventional vector-GIS and CAD approach. In GIS and CAD, the entitites of interest are characterized primarily as "points", "lines" or "polygons", with additional attributes attached. Objects are often typed by the name of the layer to which they belong, which may correspond with a single feature type, though detail may also be provided by a "type" attribute per geometry instance. In such conventional systems, identity is associated with geometry, and usually only one geometry (and scale) per feature is available.

### Tables, Objects and XML

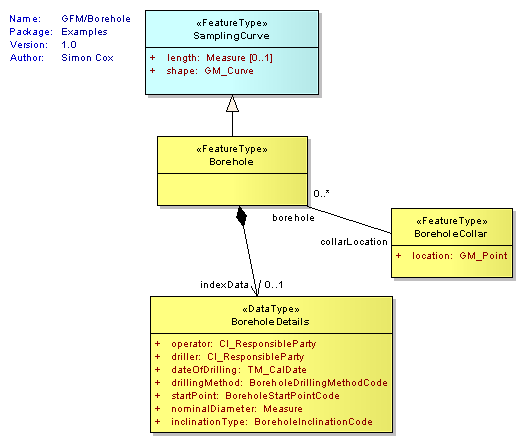
See [SchemaMapping](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/SchemaMapping).

-- [SimonCox](https://www.seegrid.csiro.au/wiki/bin/view/Main/SimonCox) - 30 Oct 2007

GML IMPLEMENTATION OF FEATURE AND PROPERTIES

LINK: <https://www.seegrid.csiro.au/wiki/AppSchemas/GmlImplementation>

### Example

Consider this extract from the [GeoSciML Borehole model](https://www.seegrid.csiro.au/wiki/bin/view/CGIModel/BoreHolesAndObservation), which is described further in [FeatureModel#Formalizing\_a\_feature\_type](https://www.seegrid.csiro.au/wiki/bin/view/AppSchemas/FeatureModel" \l "Formalizing_a_feature_type):   


Encoded following the GML rules, an XML instance of the GeoSciML[?](https://www.seegrid.csiro.au/wiki/bin/edit/AppSchemas/GeoSciML?topicparent=AppSchemas.GmlImplementation" \o "Create this topic) Borehole might look like this:

<gsml:Borehole gml:id="gcDD0214">

<gml:name codeSpace="urn:cgi:authority:CGI:GSV">gcDD0214</gml:name>

<sa:length uom="m">48.3</sa:length>

<sa:shape>

<gml:LineString gml:id="gcDD0214survey">

...

</gml:LineString>

</sa:shape>

<gsml:collarLocation>

<gsml:BoreholeCollar gml:id="gcDD0214c">

<gsml:location>

<gml:Point gml:id="gcDD0214p">

<gml:pos srsName="urn:ogc:crs:EPSG:4326">-31.939 115.832</gml:pos>

</gml:Point>

</gsml:location>

</gsml:BoreholeCollar>

</gsml:collarLocation>

<gsml:indexData>

<gsml:BoreholeDetails> ... </gsml:BoreholeDetails>

</gsml:indexData>

...

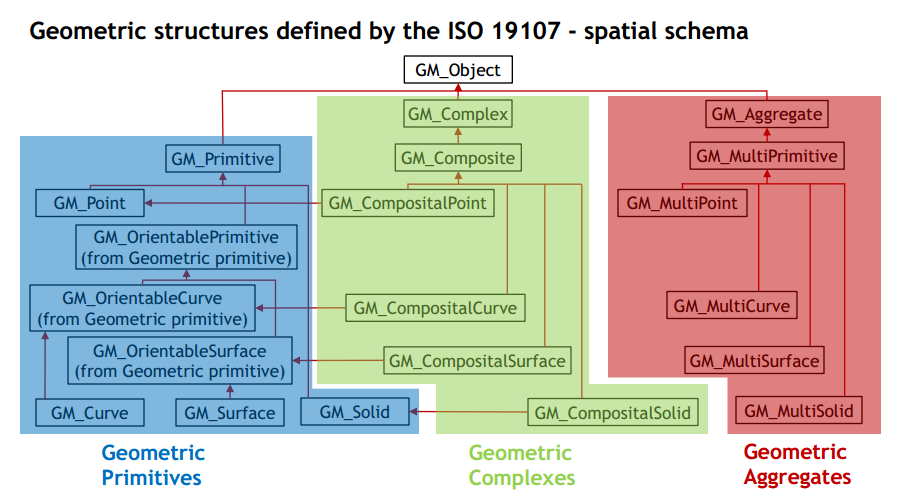
</gsml:Borehole>

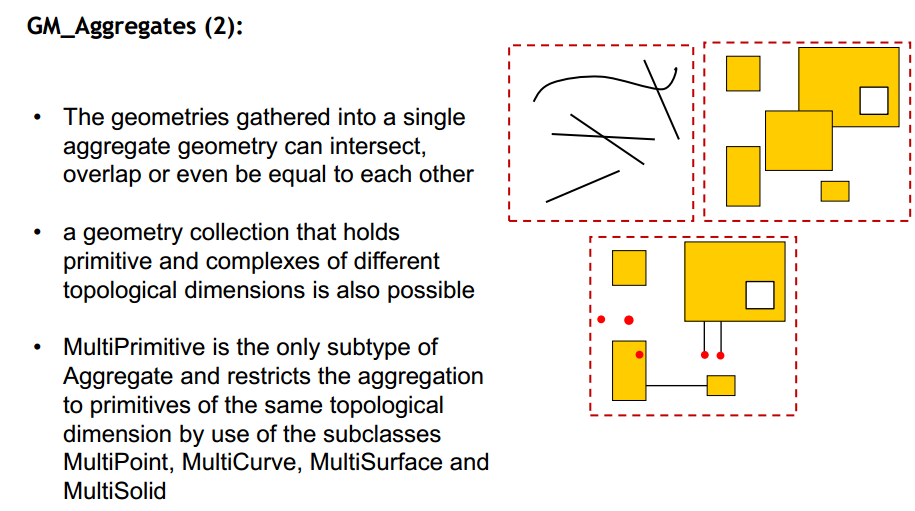
**Glossary:**

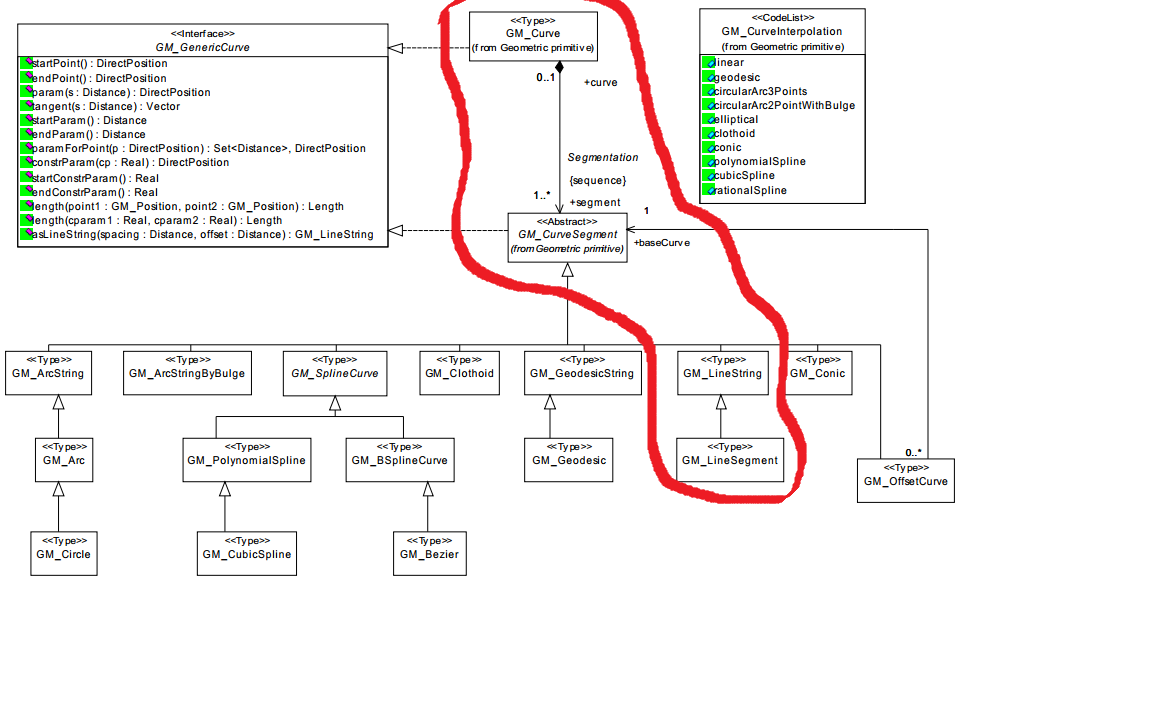
Gazetteer – geographical register (geographisches namensverzeichnis)  
Catchment – Einzugsgebiet

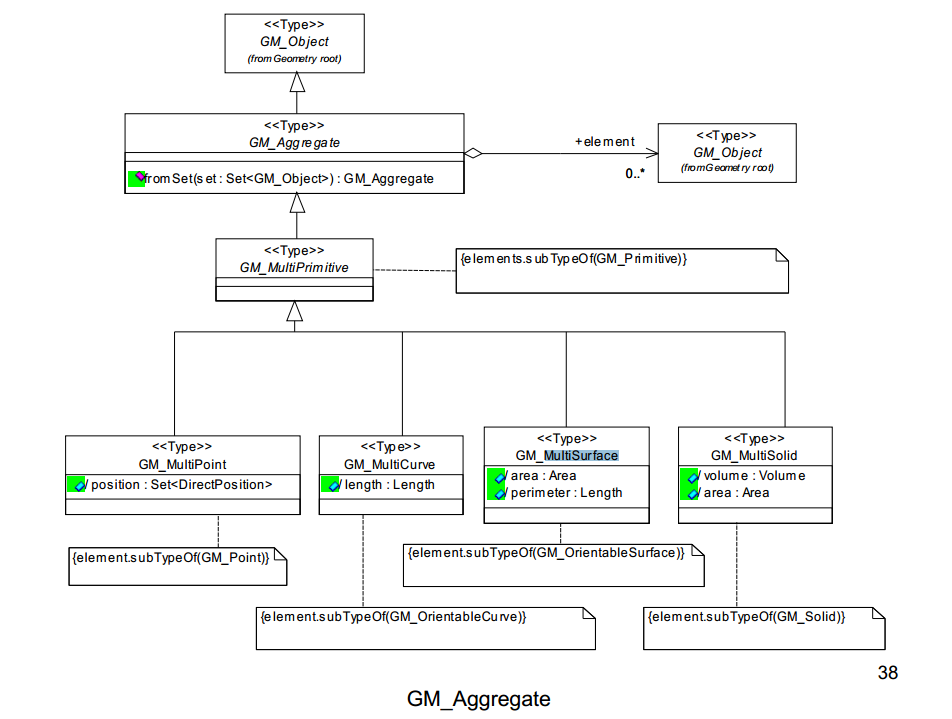
# Geo Primitives, Composition and Aggregates (ISO 19107)

****

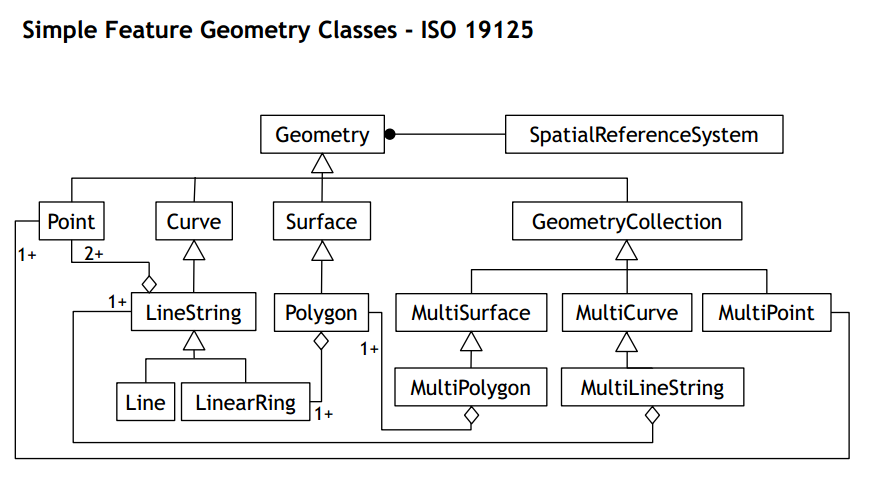




****



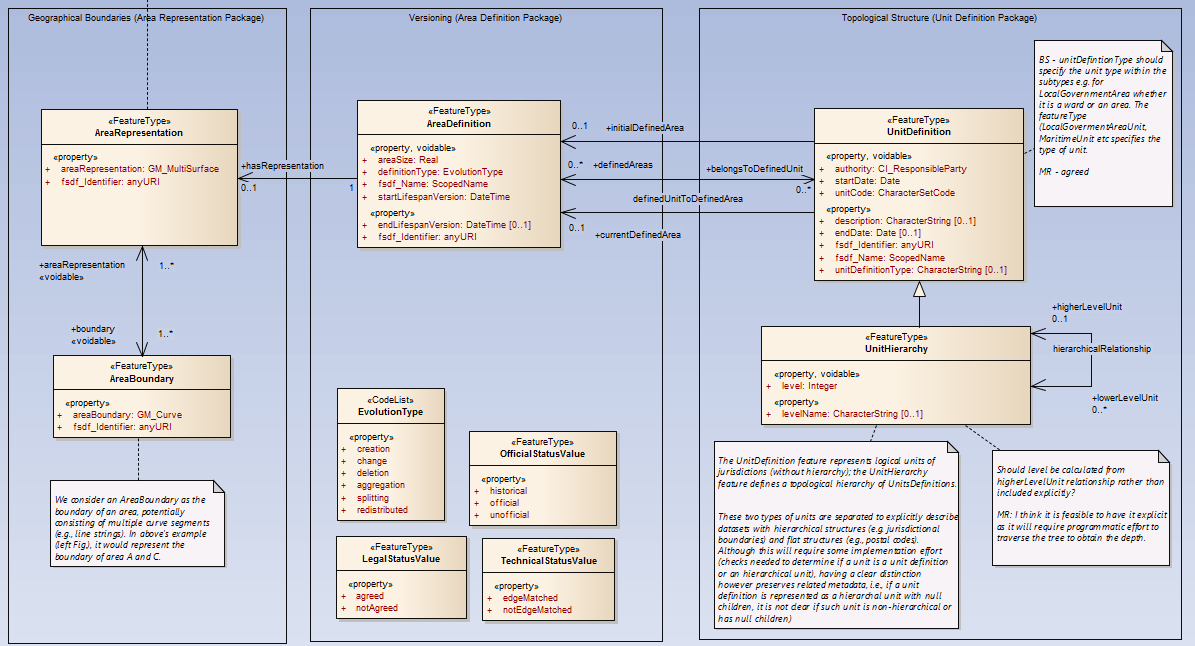
However, we are using Simple Feature Geometry (ISO 19125) (SF-1)



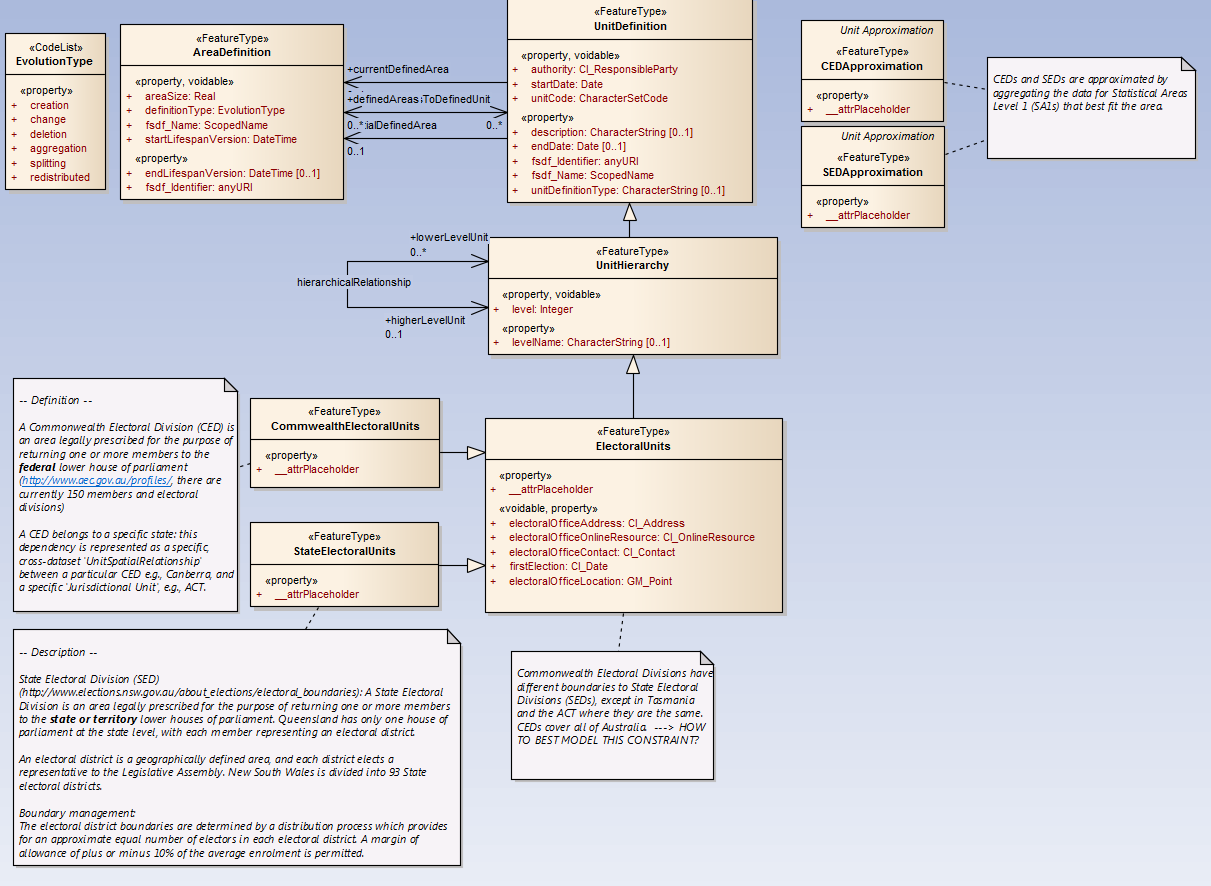
# Core Definition Model:







# Electoral Units



# Geofabric data products - an analysis of processes, descriptions and models

From a model perspective, Phase 1 data product schemata were generated manually and validated against the conceptual model. As part of Phase 2 of the Geofabric processes to augment Phase 1 products and to identify stable hydrologic concepts needed to realise the business objectives of the Geofabric are being developed. In Phase 1,the schema generation process was based on, but not driven by models. UML models for this analysis were reverse engineered from geodatabase schemas for the maintenance and product models. Using a range of Solid Ground functionality, models were imported into Enterprise Architect, cleaned (to strip out system metadata e.g. common geofabric attributes), converted to ISO compliant models, and refactored to remove the ESRI meta-model elements and extract common vocabularies. This (reverse engineering) process of generating these models and the mappings between them is the reverse sequence of steps that would be performed if end to end model driven geospatial product development were implemented. However, this reverse engineering process greatly informs the requirements for forward engineering transformations.

This report describes a gap analysis looking at the differences between actual production process to create AHGF data products, the descriptions of the process in the data product specification for data products, and the mapping between models of the product and the sources they were derived from.

* A maintenance environment to store geofabric features that will be used to produce products (the data inputs) together with models of features types in the maintenance environment;
* A means to be able to adequately describe and reference processes required to produce products from the maintenance environment (the processes).

Glossary:

First, sub-themes (statistical, electoral units) refer to different datasets. Second, themes (admin boundaries) refer to actual data products.