DGIWG 205

Defence Geospatial Information Model (DGIM)

Document type:	Standard
Document date:	28 November 2017
Edition:	2.0.0
Supersedes:	This document supersedes DGIWG 201 Ed. 1.0.0, NATO Geospatial Information Model (NGIM), 18 October 2013.
Responsible Party:	Defence Geospatial Information Working Group (DGIWG)
Audience:	This document is approved for public release and is available on the DGIWG website, http://www.dgiwg.org/dgiwg/.
Abstract:	This standard provides information on the purpose and structure of data within the Defence Geospatial Information Model (DGIM) part of the Defence Geospatial Information Framework (DGIF).
Copyright:	(C) Copyright DGIWG, some rights reserved - (CC) (By:) Attribution You are free: - to copy, distribute, display, and perform/execute the work - to make derivative works - to make commercial use of the work Under the following conditions: - (By:) Attribution. You must give the original author (DGIWG) credit. - For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder DGIWG. Your fair use and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license is available from Creative Commons <http://creativecommons.org/licenses/by/2.0/ >

Contents

List of Tables

i. Submitting organizations

Nation	Organization
Germany	Bundeswehr Geoinformation Centre (BGIC)

ii. Document point of contact

All inquiries are to be sent to secretariat@dgiwg.org.

iii. Revision history

Date	Release	Editors	Primary clauses modified	Description
2015-XX-XX	0.1	DEU	All	Initial work
2015-XX-XX	0.2	DEU	All	General rework
2015-09-14	0.3	DEU	All	General rework
2015-10-30	0.4	DEU	6	Added subchapters
2015-12-02	0.8	DEU	All	Final review
2015-12-14	0.8.1	VMST	All	Review
2016-01-08	0.9.1	DEU	All 7.2	Finalisation Added new chapter about content.
2017-04-21	0.9.2	DEU	All	Preparations for publication
2017-05-11	0.9.3	DEU	All	Preparations for Final Draft
2017-06-01	0.9.4	DEU	All	Final Draft
2017-09-29	0.9.5	DEU	All	Comment Resolution

iv. Future work

As the DGIF programme of work continues and business and technical process matures, it is expected that the DGIM will evolve and this specification reviewed and updated accordingly.

Introduction

This Standard has been developed as part of the Defence Geospatial Information Framework (DGIF) suite of standards.

The DGIM is a DGIF-wide logical model for geospatial data that is technology neutral. As a Platform Independent Model (PIM) it simply determines the syntactic structure.

The Defence Geospatial Feature Concept dictionary (DGFCD) provides all Concepts required by DGIM. The initial DGFCD was a profile of the DGIWG Feature Data Dictionary (DFDD).

The Defence Geospatial Real World Index (DGRWI) provides an entry point into the DGIM by providing indexing from names of Real World Phenomena to their implementation in the DGIM.

1 Scope

The Defence Geospatial Information Model (DGIM) specifies a logical model for geospatial data for the defence and intelligence community that is technology neutral. As a Platform Independent Model (PIM) it determines the syntactic structure of the semantics used in the DGIF. From it, using Model Driven Architecture (MDA) techniques, technology-tied Platform Specific Models (PSM) may be automatically derived and directly employed in system development.

The DGIM conforms to ISO 19109, Geographic information – Rules for application schema, and its conceptual schema. It integrates conceptual schemas from multiple ISO 19100-series standards for geospatial information modelling.

The DGIM ensures that there is a clear, complete, and internally-consistent DGIF logical geospatial data model that may be used to derive system-specific implementation models/schemas in a rigorous manner – this ensures that data integrity is preserved when geospatial data is exchanged between different system implementations within the DGIF. It also reduces the cost of evolving system-specific implementations to meet evolving system, mission and customer requirements while promoting data agility.

The DGIM leverages and integrates geospatial information modelling practices from multiple community models (e.g., MGCP, AIXM, ENC, AML, and others) whose data are used and exchanged by DGIF component systems.

ISO 19101, Geographic information – Reference Model, defines a feature as an abstraction of real world phenomena. Such abstractions may be represented in information systems using a variety of spatial modelling methods, including representations such as vectors, grids and images. The DGIM supports this breath of geometric representations for “feature data” in the DGIF. The DGIM also supports modelling entities that may represent other geospatially-located information that does not correspond to “real world phenomena”. Unless otherwise specifically stated, the terms feature and (modelling) entity are used interchangeably in this standard.

Individual items of feature and/or attribute information that are used in the DGIM are specified by the Defence Geospatial Feature Concept Dictionary (DGFCD). Through the DGFCD the DGIM draws upon recognized content standards, specifications and profiles from both the military (e.g., DGIWG) and civilian sectors (e.g., IHO, ICAO/EUROCONTROL, WMO).

Real world phenomena are described in the Defence Geospatial Real World Object Index (DGRWI) which can be used as an entry point to the DGIM.

Information traceability is established from concepts in the DGIM to their specification in the DGFCD, and from there back to appropriate authoritative concept sources, where possible, to maximize semantic integrity when geospatial data is exchanged between DGIF-based and external systems. The DGIM, the DGFCD and the DGRWI taken together answer the information exchange questions of “*what do we mean?*” and “*how do we represent it?*”

2 Conformance

The DGIM conforms to ISO 19109, *Geographic information – Rules for application schema,* and its conceptual schema. It integrates conceptual schemas from multiple ISO 19100-series standards for geospatial information modelling.

3 Normative references

The documents listed in Table 1 are indispensable to understanding and using this standard. For dated references, only the cited edition or version applies. For undated references, the latest edition or version of the referenced document (including any amendments) applies.

For information purposes Table 2 shows references to the predecessor and starting point, respectively, of this standard, the NATO Geospatial Information Model in its version #1.

Table 1: Normative References

Standard or Specification
ISO/TS 19103:2005 - Geographic information – Conceptual schema language
ISO 19107:2003 - Geographic information – Spatial schema
ISO 19109:2005 - Geographic information – Rules for application schema
ISO 19110:2005 - Geographic information -- Methodology for feature cataloguing
ISO 19111:2003 - Geographic information – Spatial referencing by coordinates
ISO 19112:2003 - Geographic information – Spatial referencing by geographic identifiers
ISO 19115:2003 - Geographic information – Metadata
ISO 19123:2005 - Geographic information – Schema for coverage geometry and functions
ISO 19135:2005 - Geographic information - Procedures for item registration
ISO/IEC 19505-1:2012 - Information technology - Object Management Group Unified Modeling Language (OMG UML) - Part 1: Infrastructure".
ISO/IEC 19505-2:2012 - Information technology -- Object Management Group Unified Modeling Language (OMG UML) - Part 2: Superstructure
ISO/IEC 19507:2012 - Information technology -- Object Management Group Object Constraint Language (OCL)
DGIWG - 206 - Defence Geospatial Feature Concept Dictionary (DGFCD)
DGIWG - 207 - Defence Geospatial Real World Object Index (DGRWI)
DGWIG - 208 - DGIF Exchange Formats for Vector Data - Part-1 GML
DGIWG - 114 - DGIWG Metadata Foundation

Table 2: Informative References

Standard or Specification
DGIWG – 201 - NATO Geospatial Information Model (NGIM) https://portal.dgiwg.org/files/?artifact_id=8622&format=pdf
DGIWG – 202 - NATO Geospatial Entity Catalogue (NGEC) - Normative Content https://portal.dgiwg.org/files/?artifact_id=8624&format=pdf
DGIWG – 203 - NATO Geospatial Feature Concept Dictionary (NGFCD) https://portal.dgiwg.org/files/?artifact_id=8625&format=pdf
DGIWG – 204 - NATO Geospatial Real World Object Index (NGRWI) https://portal.dgiwg.org/files/?artifact_id=8626&format=pdf

4 Terms, definitions, and abbreviations

4.1 Definitions

Terms and definitions specific to this standard are given in Table 3.

Table 3: Definitions applicable to this standard

Term	Definition
Entity	A modelling class that may represent either a feature or other geospatially-located information.
Entity Association	A relationship that links instances of one entity type with instances of the same or a different entity type.
Entity Attribute	A characteristic of an entity.
Feature	An abstraction of real world phenomena.
DGIM	A structured collection of feature information (features, attributes, associations, and ancillary data) whose metamodel conforms to the general feature model as specified in ISO 19109.

4.2 Abbreviations

The acronyms that are used in this standard are specified in the following list.

AIXM	Aeronautical Information Exchange Model
AML	Additional Military Layers
DGFCD	Defence Geospatial Feature Concept Dictionary
DGIF	Defence Geospatial Information Framework
DGIM	Defence Geospatial Information Model
DGIWG	Defence Geospatial Information Working Group
DGRWI	Defence Geospatial Real World Object Index
DMF	DGIWG Metadata Foundation
DTM	Defence Topographic Map
ENC	Electronic Navigational Chart
EUROCONTROL	European Organisation for the Safety of Air Navigation
FACC	Feature and Attribute Coding Catalogue
GML	Geography Markup Language
ICAO	International Civil Aviation Organization
IEC	International Electrotechnical Commission
IHO	International Hydrographic Organization
ISO	International Organization for Standardization

JGSWG	Joint Geospatial Standards Working Group
JMCDM	Joint METOC Conceptual Data Model
KML	Keyhole Markup language
MDA	Model Driven Architecture
METOC	Meteorology and Oceanographic
MGCP	Multinational Geospatial Coproduction Program
MGID	Military Geographic Information & Documentation
MIME	Multipurpose Internet Mail Extension
NATO	North Atlantic Treaty Organization
NGEC	Defence Geospatial Entity Catalogue
OASIS	Org. for the Advancement of Structured Information Standards
OCL	Object Constraint Language
OGC	Open Geospatial Consortium
OMG	Object Management Group
PIM	Platform Independent Model
PSM	Platform Specific Models
SI	International System of Units
SKOS	Simple Knowledge Organization System
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WMO	World Meteorological Organization
XSD	XML Schema Document

5 Logical Structure

5.1 Conceptual Metamodel

The DGIM's conceptual metamodel conforms to the general feature model as specified in ISO 19109, *Geographic information – Rules for application schema*. This metamodel is described in chapter 9. The ISO 19109 metamodel has also been partly simplified and additional classes are added to support the organization and presentation of DGIM content.

As necessary, the metamodel specified in ISO 19109 has been extended based on the object modelling component of *OMG Unified Modeling Language (OMG UML), Superstructure*, Version 2.2, September 2009. In particular, the class and datatype metamodels from the Kernel package have been used for extension guidance.

5.2 Logical Metamodel

In general the metamodel is organized as follows.

There are five basic categories of elements: **Feature Types**, **Types**, **Data Types**, **Unions and Enumerations**. Additional elements like **Bundles** and **Leaves** support the management of the DGIM.

Feature Types and **Types** are characterized by properties called **Attributes** and **Roles**. **Enumerations** contain **Enums** as values.

Types and **Feature Types** can be linked using **Associations** with specific properties called **Association Roles**.

Generalisations between **Types** or **Feature Types** can be used to inherit information and to build hierarchical dependencies.

Elements are linked to those in the DGFCD (**Feature Concepts**, **Attribute Concepts**, **Role Concepts** and **Attribute Value Concepts**) to gain precise meaning using **Definition**.

Combinations of elements defining a specific meaning can be designated by defining **Real World Objects** in the DGRWI using **Representation** link.

Each Element owns basic information needed for maintenance and management following the principles of item registration by ISO 19135 – Geographic information - Procedures for item registration.

5.2.1 General Model Specifications

Each model element in DGIM requires specific information as following:

1. **(Class) Name:** The class name represents the alphaCode, a unique alphanumeric value that may be used to designate this concept for the purposes of data interchange between nations, organisations and within NATO. The alphaCode is also called primary code and represents a human readable name that can be longer than 25 characters. It should be use as primary identification though since information systems may be not able to use identifiers of this length, the 531Code may be used as defined in the DGFCD.
2. **Alias:** The alias represents the 531Code, a unique alphanumeric value that may be used to designate this concept for the purposes of data interchange within the DGIF

in technology-specific limited circumstances. The 531Code is also called secondary code and should only be used in the case the alphaCode cannot be used in the system environment.

3. **Status:** The status of the element following ISO19135 specifications. Allowed values are *valid*, *notValid*, *retired*, *superseded*. Only valid items are part of the current model.
4. **DateAccepted:** The date when the element was accepted to be a valid element with the DGIM.
5. **DateRetired:** The date when the element was retired from the DGIM and became *retired* or *superseded*.
6. **SecurityClassification:** A specific security specification should be allowed to use. This model part is not used at the moment.

5.2.2 Multiplicity

The information defining whether a property is mandatory or optional is documented by the multiplicity as shown in Table 4:

Table 4: Options for defining the multiplicity

Option	UML	Meaning
Optional	[0..	*No value is necessary*
Mandatory	[1..	A least one value has to be provided.
Open	[..*]	Indefinitely many value can be provided
Closed Interval	[1..3]	At least one but not more than three values shall be provided.
Exactly One	[1]	Exactly one value has to be provided.
Exactly One		Exactly one value has to be provided. If "[1]" is meant the brackets can be removed.

Depending on the type of element, additional information may be specified. This additional information is specified in the following subsections of this standard. Technical information, for example for the creation of GML, including format and syntax can be found in DGIWG - 208 - Defence Geospatial Information Framework - Encoding Specification - Part-1: GML.

5.2.3 Bundle

A **bundle** is basically[1] a package consisting of other packages. It is named and is defined by a documentation and optional a short description.

Bundles are created by themes based on the content.

5.2.4 Leaf

A **leaf** is a package without any more sub packages that comprises of DGIM elements. It is named and is defined by a documentation and optional a short description.

[1] Occasionally, also featureTypes may reside in the bundle hierarchy level if the creation of a theme does not seen necessary, e .g. because there is only one element.

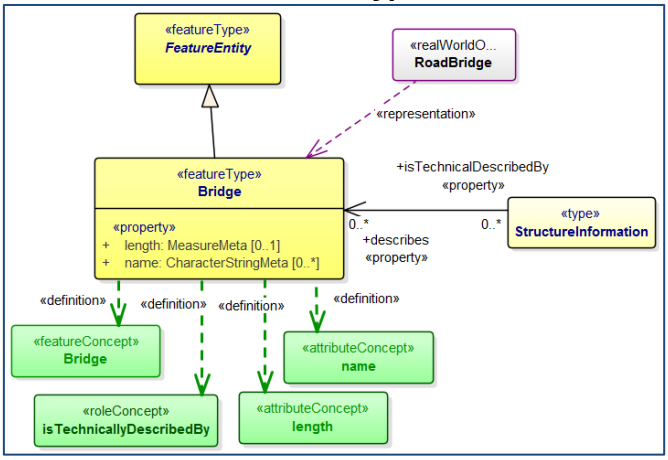
Leaves are created by themes based on the content.

5.2.5 FeatureType

A **featureType** equals a feature as described in ISO19110, an “abstraction of real world phenomena”, meaning an entity with a geospatial relation.

FeatureTypes build the core elements for describing content.

Several **taggedValues** give more information on the actual **featureType**.



The **taggedValue geometry** specifies the different geometries allowed for the **featureType**. Some **featureTypes** may only be established by specific geometries, for example a *SpotElevation* is only represented by a *point* while a *River* may be represented as *curve* or *surface*.

The **taggedValue profiles** allows the assignment to a specific profile and thus to a data content and data product specification. Within this **taggedValue** it is possible to restrict the geometry for the **featureType** within the profile even more, for example, a *River* is only allowed as *curve* in the product specification for 1:50,000 Defence Topographic Map (DTM50).

FeatureTypes are connected to other **featureTypes** and **types** by **associations** describing relations to those. **Generalizations** allow the creation of subtyping as specified in UML.

They are connected to **featureConcepts** by **definition** to link the **featureType** with its name, definition and description maintained in the DGFCD.

For the same reason **featureTypes** are connected by **definition** to all **attributeConcepts** and all **roleConcepts** that are defining attributes or roles used by the **featureTypes**.

A **featureType** is connected from **realWorldObject** (DGRWI) it is representing by **representation**.

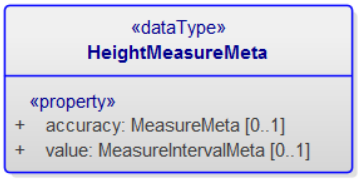
5.2.6 Type

A **type** is an information entity that has no geometry on its own, for example to describe general information as *ReligiousInformation*. It behaves as **featureTypes** except those parts referring to geometry.

5.2.7 Data Type

A **dataType** defines the structure of a property and specifies the general nature of the content. Simple **dataTypes** are real or string. In DGIM simple types are not explicitly modelled but derived from ISO19103.

DataTypes are used to construct the data type triple (see chapter 6.3.5) and to establish complex types that consist of several (sub)properties, for example *HeightMeasureMeta*, that comprises not only of the actual value but



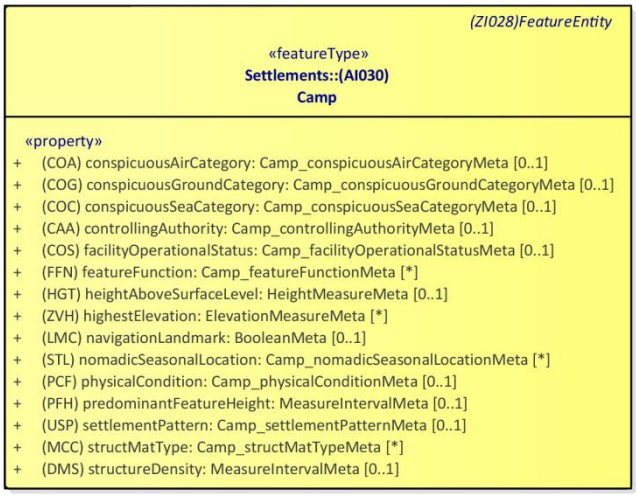
also a property to store the accuracy.

To explain the nature of the **dataType** a **description** can be populated.

5.2.8 Union

A **union** is a special type of **dataType**. All **properties** stored in a union can be understood as alternatives, for example the union *MeasureReason* is representing an actual measured value or an ONINA (Other, No Information, Not Applicable).

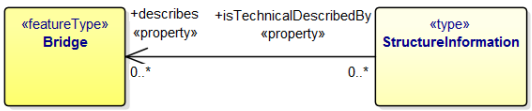
5.2.9 Attribute



Attributes are properties of elements. They describe specific characteristics on these elements. **Attributes** have a specific **dataType**. They have a **multiplicity** as stated in UML defining how often an **attribute** appears or how many values are allowed. In the case of an indefinite multiplicity (*) the **taggedValue maxOccurs** allows to limit the actual number of values to a definite value[2]. A multiplicity starting with “1” defines a mandatory property for the element. The **taggedValue profiles** allow the assignment to a specific profile and therefore to a data content and data product specification. Within this **taggedValue** it is possible to restrict the **maxOccurs** for this specific **attribute** on the **featureType** within the profile even more, for example the name of a *River* is only allowed for 2 times in the profile *DTM50* even if the general **maxOccur** is set to a higher value.

5.2.10 Associations

Associations represent the relation between two elements (**types** or **featureTypes**). Compositions and aggregations as defined in UML are included. **Associations** shall not be used to describe spatial relations between **featureTypes** since those relations between features can be derived from spatial analysis of actual data in a specific product/implementation. **Associations** should be created with care since they have massive effect on schema derivations. More information on the usage of **associations** is given in chapter 6.5. **Associations** are accompanied by **associationRoles**.

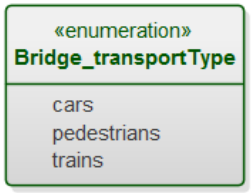


5.2.11 Role

Roles define the relation between elements via **associations**. Details can be found in the UML specification. Similar to **attributes** the **taggedValues maxOccurs** and **profiles** can be set.

[2] Note that a default value of 3 is used in the derivation of schemas using the software ShapeChange, even if maxOccurs is empty.

5.2.12 Enumeration



An **enumeration** represents the range of a property in the shape of a so called pick list. Only one or more of the values (**enums**) within this list is an allowed value for the **property**.

An **enumeration** itself can be part of a profile. The **taggedValue profiles** have to be populated properly. They are linked to all **attributeValueConcepts** that define the **enums** in the **enumeration** via **definition**.

5.2.13 Enum

Enums represent the values in an **enumeration**. They are subject to the **taggedValue profiles** as described above.

5.2.14 Codelist

Codelists are external maintained and referenced lists of values. They are currently not used in DGIF.

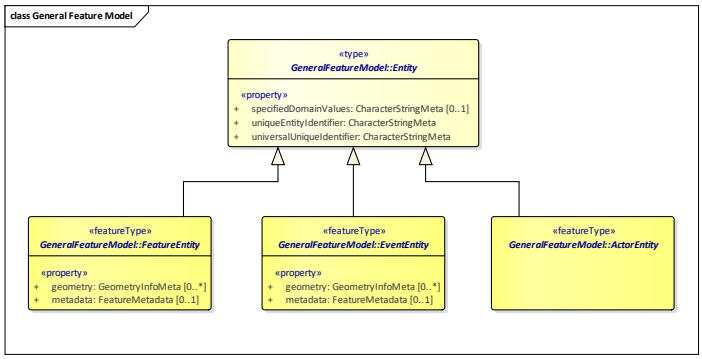
6 Modelling Patterns and Principles

This chapter describes basic principles and patterns that are used in the modelling work of DGIF.

6.1 General Feature Model

The model is based on three different (abstract) entities that inherit from the root class **Entity**.

Each **Entity** owns three attributes:



specifiedDomainValues: One or more intended attribute domain values for one or more enumeration or codelist attributes that are not currently valid members of their respective attribute ranges.(This attribute allows storing values for enumerations where “other” was selected).

uniqueEntityIdentifier: The globally unique and persistent identifier of an entity (for example: feature or event) instance as specified by a Uniform Resource Name (URN) in accordance with the Internet Engineering Task Force (IETF) RFC2396 and RFC2141.

universalUniqueIdentifier: The Universal Unique Identifier (UUID) that is assigned to a feature in order to uniquely identify it for the purpose of maintaining relationships between features (relational database) and version control. It is defined in ITU-T Rec. X.667 and ISO/IEC 9834-8, respectively.

6.1.1 Feature Entity

An **Entity** that describes a feature as defined in ISO 19110 (add definition). For example *Bridge*, *Building*, *Harbour*. All **featureTypes** inherit from **featureEntity**.

FeatureEntity owns two attributes:

The **attribute** *geometry* is of type *GeometryInfoMeta*. This attribute provides the creation of geometry for a **featureType**. The *GeometryInfoMeta* class is described in chapter 6.1.3.

The **attribute** *metadata* is of type *FeatureMetadata* and provides the link to metadata on feature level. The class is described in chapter 6.4.1.

6.1.2 Event Entity

An Event Entity is an **Entity** that describes an event of geospatial nature. Currently only the *SignificantEvent* is used to cover religious or conflict events.

The **attribute** *geometry* is of type *GeometryInfoMeta*. This attribute provides the creation of geometry for a **featureType**. The *GeometryInfoMeta* class is described in chapter 6.1.3.

The **attribute** *metadata* is of type *FeatureMetadata* and provides the link to metadata on feature level. The class is described in chapter 6.4.1.

6.1.3 Actor Entity

An Actor Entity is an abstract modelling **Entity** that is a superclass for actor types, which are representations of intentional (that is, purposeful) entities that act or have the capability of acting as a participant in an event. An actor may control resources or direct activity, and may do so by holding an organizational role. Actor entities include persons, social groups, organisations, and, in some contexts, automata. Collective actors (for example: organizations) may have a distributed rather than a point location.

6.2 Geometry Model

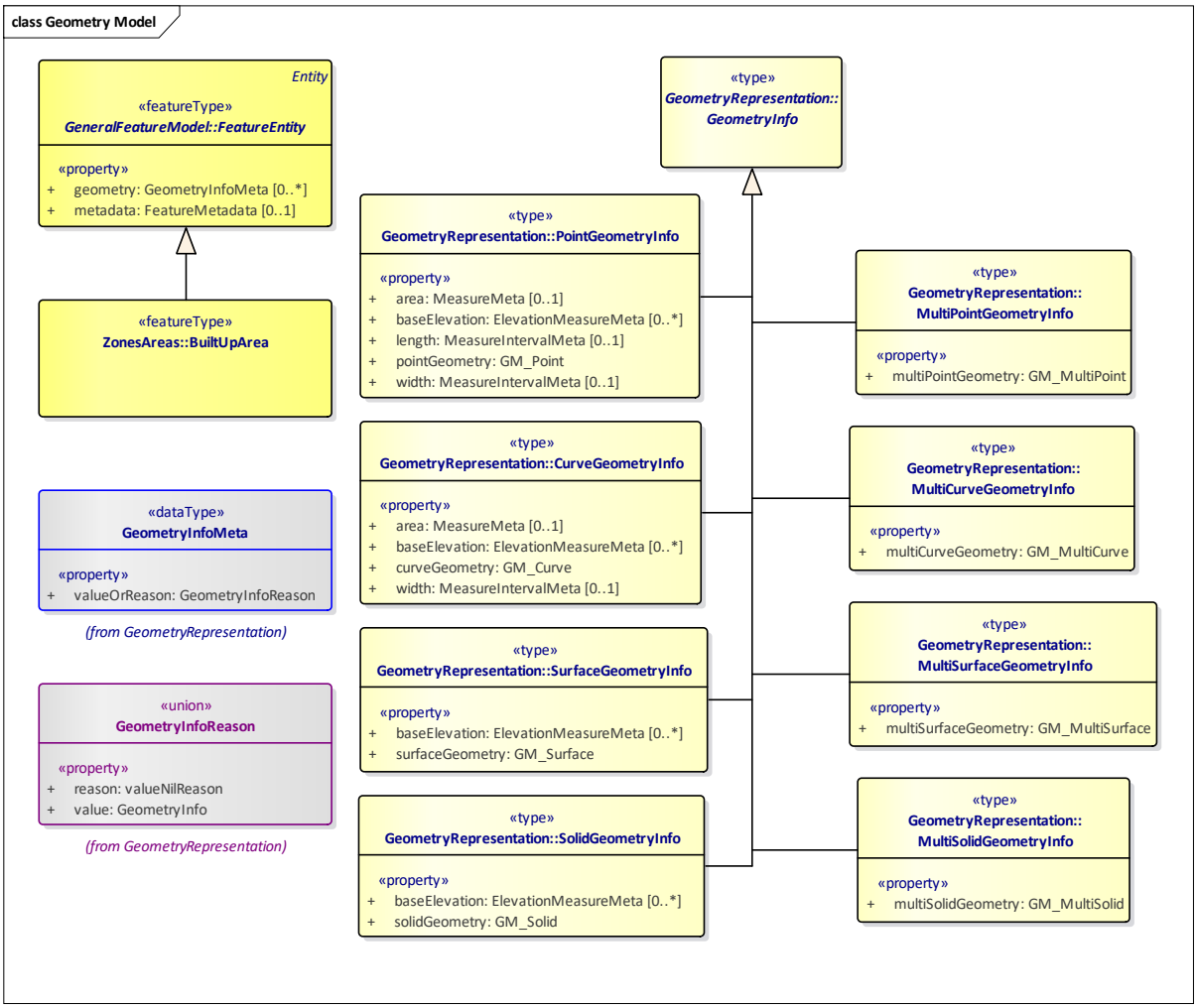
The Geometry model is based on ISO 19107.

For DGIF, the following geometries are allowed: *Point*, *multipoint*, *curve*, *multicurve*, *surface*, *multisurface*, *solid* or *multisolid*. The detailed description of these geometries can be found in ISO 19107.

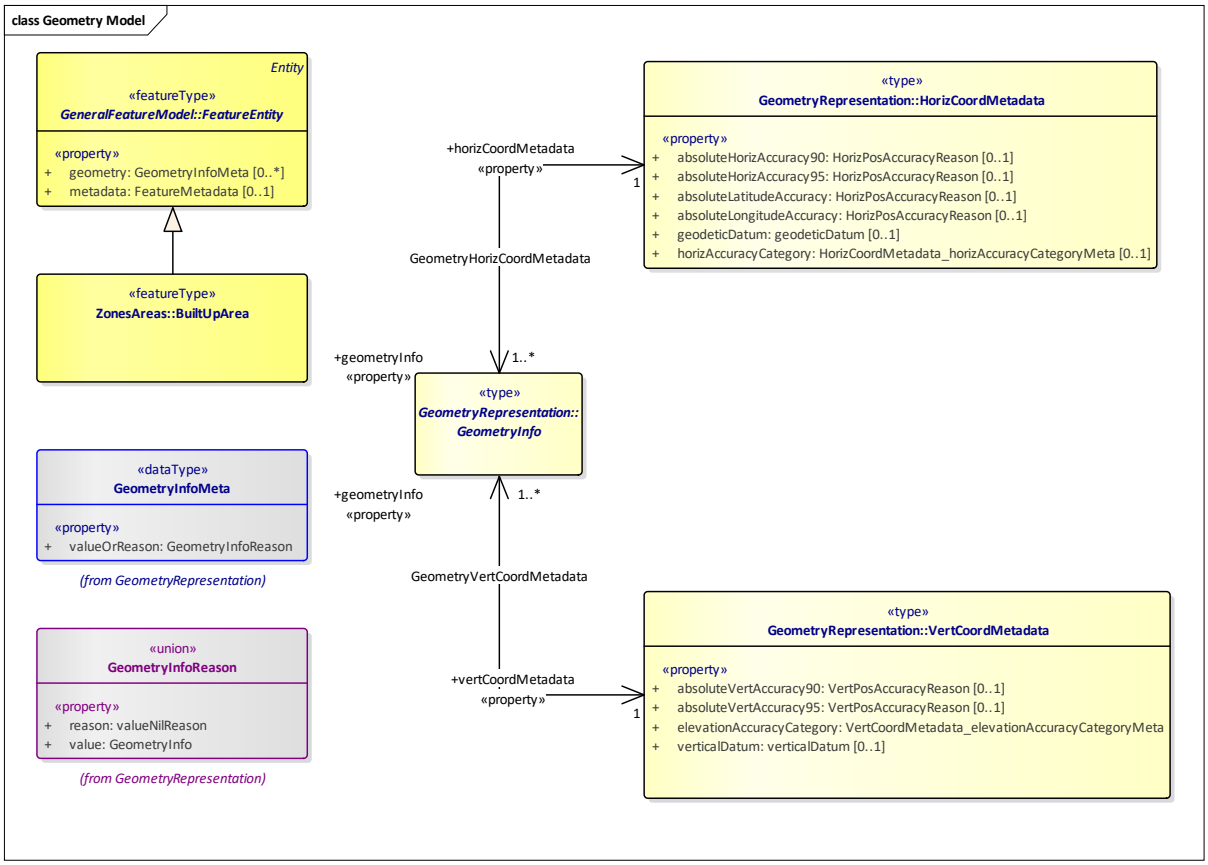
Each *geometry* has an attribute linking the corresponding ISO19107 elements and specific geometric attributes. The attributes enable the storage of geometric information that cannot be measured in the geometry itself. Every *point* owns length, width, area and base elevation. Every *curve* owns width, area and base elevation since the length can be derived from the geometry itself. Every *area* owns only base elevation.

Multigeometries do not own geometric attributes since they consist of many different possible entities.

Each **featureType** in DGIM can be of any geometry. The attribute *geometry* is inherited from *featureEntity* with an indefinite multiplicity (*). Any restriction on the geometry can be done in the derivation process for product specific models. The taggedValue geometry restricts the possible geometries. Additionally, for any profile additional restrictions are possible. For example, *BuiltUpArea* is restricted to only *point*, *curve* or *surface* in general but may be restricted to only *point* in a very low scale profile/product.



Associated with *GeometryInfo* is metadata describing horizontal accuracy and vertical accuracy.



6.3 Data types

In DGIM three different classes of data types are used.

6.3.1 Simple Types

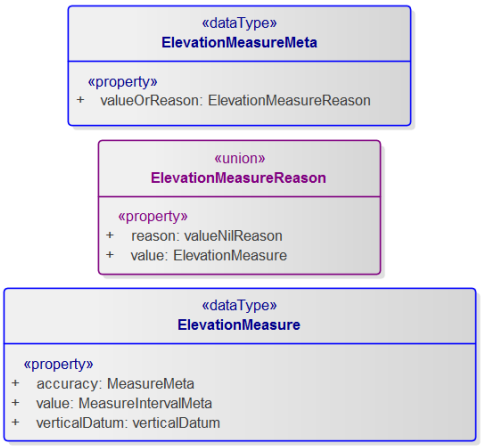
Simple Data Types represent a single value like a text or number. They are mostly derived from ISO19103. Following data types are used:

1. CharacterString: free text field, e.g. “This is an example”
2. Integer: zero and natural numbers for countable values, e.g. 234 or 9721
3. IntegerInterval: an interval consisting of natural numbers, e.g. [0;21] or [4;*)
4. Measure: a floating point with a unit of measure used for all values that can be measured, e.g. 15.8 metres, 300.75 km/h
5. MeasureInterval: an interval of floating points with a unit of measure, e.g. [20.2;25.12] metres, (1;*) nautical miles.
6. Boolean: a value representing ‘true’ or ‘false’

6.3.2 Complex Types

Complex Types represent values that are accompanied by additional information such as accuracy, referenceSystem or similar. They are created on a case by case basis.

An example for such a **dataType** is the **dataType** *ElevationMeasure* is a complex type that consists not only of the actual *value* but of a property stating the *accuracy* and a property stating the used *verticalDatum*.



6.3.3 Enumerations

Enumerations represent a list of allowed values, e.g. “blue, red, yellow, white”. They are created individually for each featureType/attribute relation.

6.3.4 Relations to DGFCD attribute data types

Data Types are defined differently in the DGFCD since the dictionary level is on a higher level of abstraction. Table 5 shows the relations and has to be obeyed.

Table 5: Relations of attribute data types between DGFCD and DGIM

DGFCD	DGIM
Boolean	BooleanMeta
Count	IntegerMeta
Integer	MeasureMeta
Key	CharacterStringMeta
Real, Real180 etc.	MeasureMeta
RealInterval	MeasureIntervalMeta
StructuredText	CharacterStringMeta
Text, Text64 etc.	CharacterStringMeta
Enumeration	[FeatureType_attribute]Meta

6.3.5 Data Type Triple

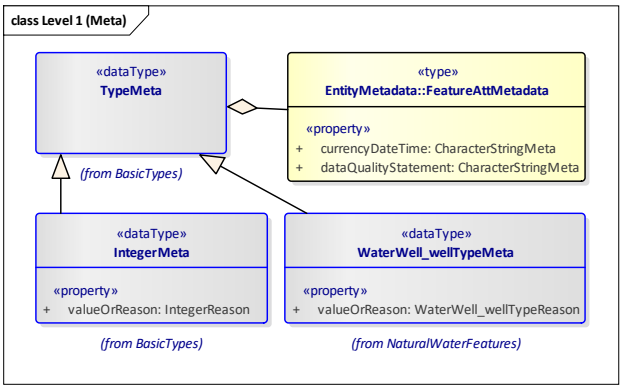
Data Types represent the range of attributes. Usually data types are represented by single types such as characterString, real, measure or similar.

In DGIF a so called Data Type Triple is introduced, a complex structure of data types and unions to allow the modelling of attribute metadata and the inclusion of default nil reasons in the model. The nil reasons are called ONINAS (other, noInformation and notApplicable) and represent the absence of values for an attribute.

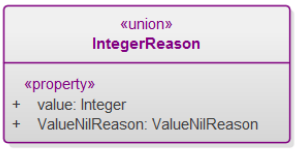
6.3.5.1 Level Hierarchy

The triple consists of three levels of data types which are also listed in Table 6:

Level 1 (Meta): Each **attribute's** type is a **data type** with a suffix "Meta". This data type, for example *CharacterStringMeta* or *IntegerMeta* is stored in the Foundation Package within DGIM. These datatypes are created to allow metadata for attributes. For example, it is possible by using this pattern to attach a dataQualityStatement to an attribute that may differ from the metadata of the overall featureType (for example, because it was derived from a different source). The level 1 structure includes an inheritance from the class *typeMeta* which is connected to attribute metadata (*FeatureAttMetadata*).



Level 2 (Reason): The second level is called reason and established by a union data type that has the suffix "Reason". The union type allows in the physical model (implementation) to choose if an actual value exists and to enter it or, if it does not exist, to enter a standard value to express the null reason.



Level 3 (Value): The third level at last is the actual value. For simple types mostly a type derived from ISO19103 such as integer or measure. In the case of an enumeration data type this is the actual enumeration stored in the leave.

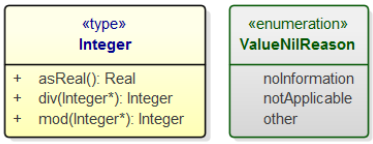


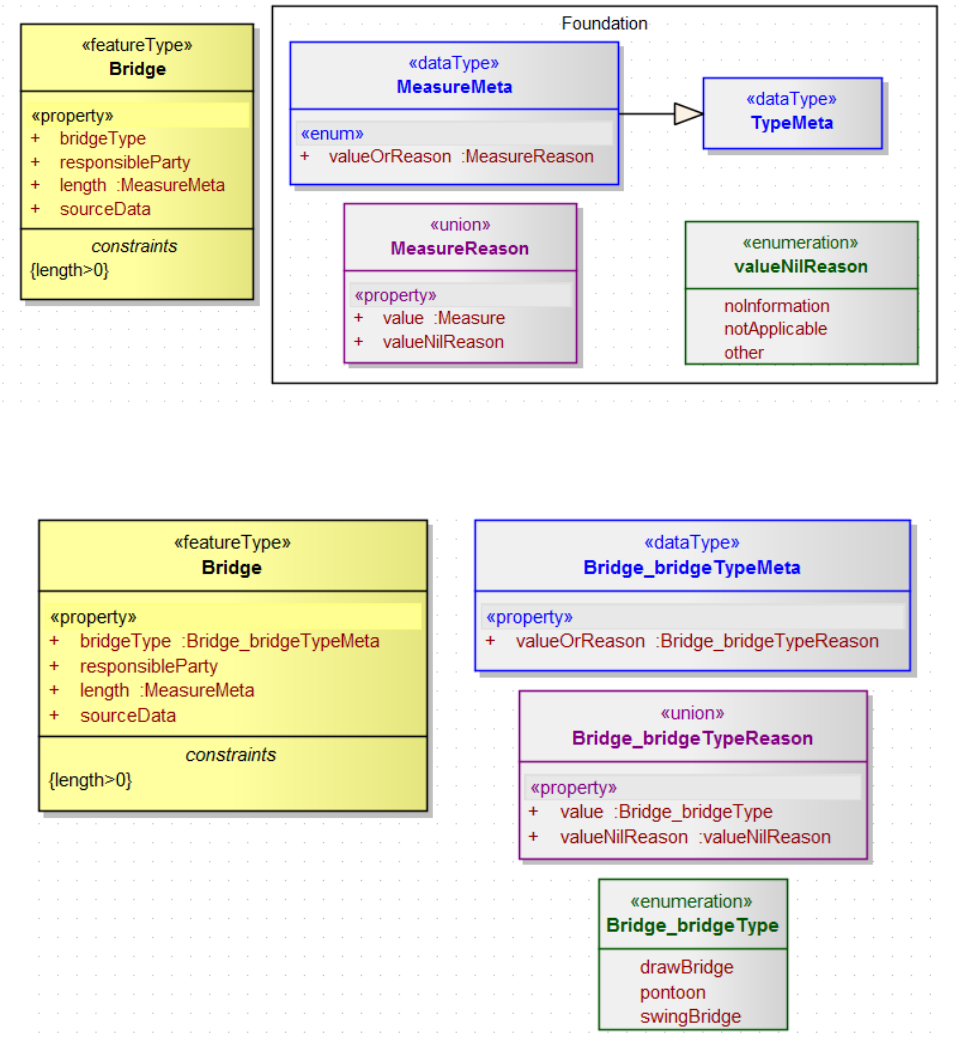
Table 6: Data types of the ONINAS level hierarchy

3 levels of data types				
Type of values	**Meta:** Including Metadata	**Reason:** Incuding Oninas	**Additional Construct**	**Basic Data Type**
Number	IntegerMeta	IntegerReason		Integer (ISO 19103)
Floating	RealMeta	RealReason		Real (ISO 19103)
Measurement	MeasureMeta	MeasureReason		Measure (ISO 19103)
Interval of Measurement	MeasureIntervalMeta	MeasureInterval-Reason	MeasureIntervalUnion and MeasureInterval	Measure (ISO 19103)
Interval of Numbers	IntegerIntervalMeta	IntegerInterval-Reason	IntegerIntervalUnion and IntegerInterval	Integer (ISO 19103)
Text	CharacterStringMeta	Character-StringReason		CharacterString (ISO 19103)
Boolean	BooleanMeta	BooleanReason		Boolean (ISO 19103)

All data types of ISO 19103 and ISO 19115 are modelled once in the package foundation and can be reused every time when needed. Hence for such general data types as characterString, real or measure, the attribute has just to be linked.

For all enumeration types the whole triple has to be created individually and is stored in the thematic leaves.

6.3.5.2 Triple Examples

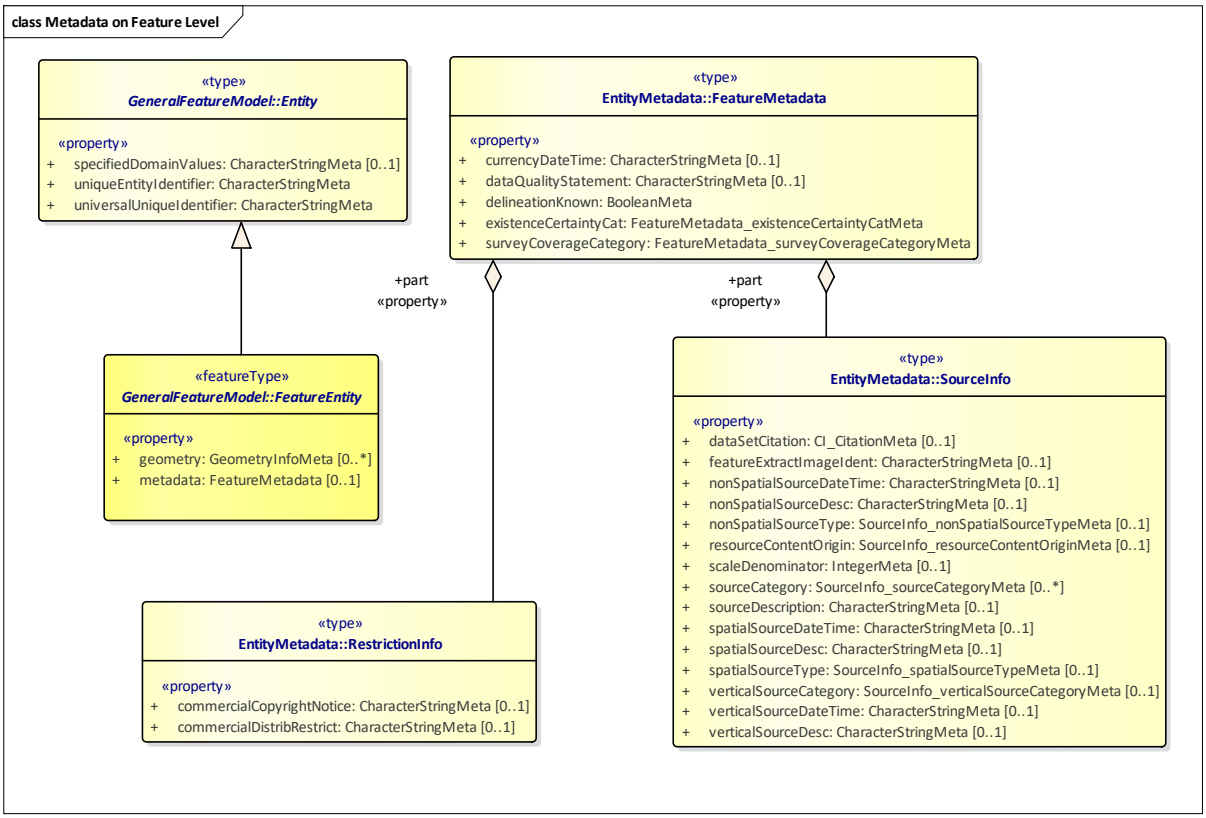


6.4 Metadata

Metadata for geospatial vector data is generally stored on three levels, for data sets, for features and for attributes. The first level, for whole datasets, is not covered by this specification. A dataset should be accompanied by a DGIWG Metadata Foundation (DMF) compliant metadata set.

6.4.1 Metadata on Feature Level

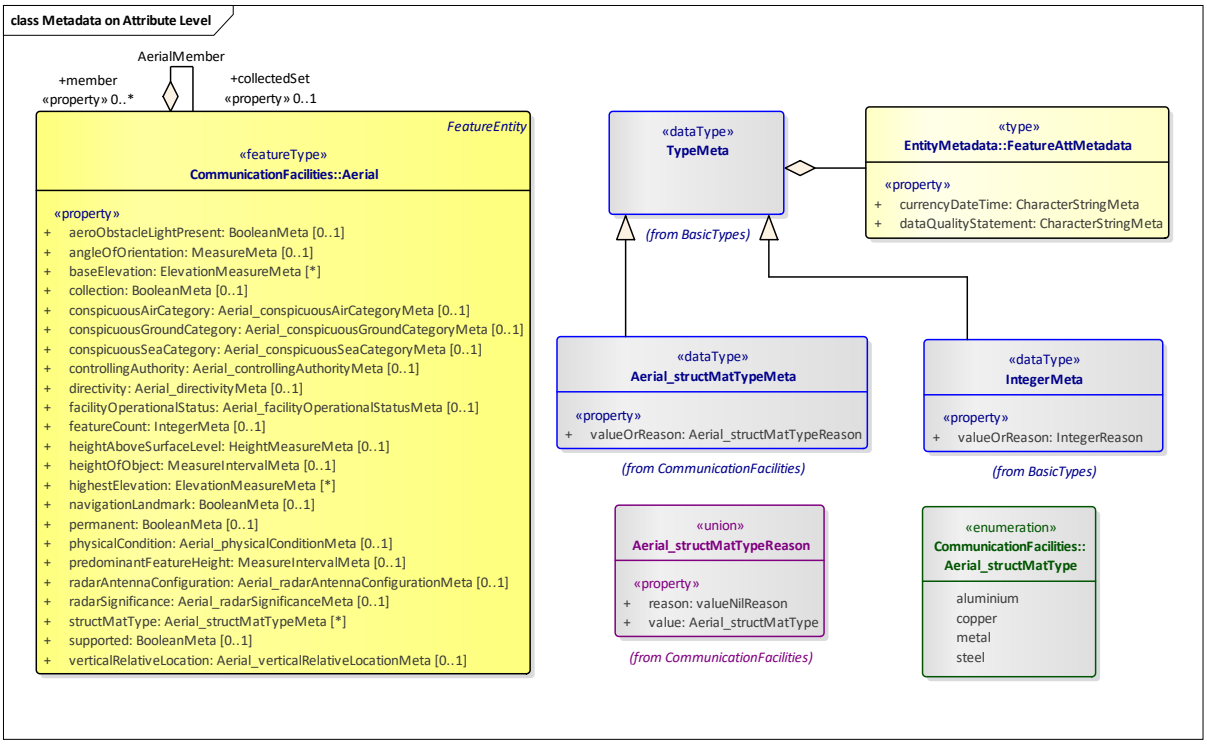
The second level is for features and is described by *FeatureMetadata* and its associated classes as can be seen in the figure. Additional metadata about accuracy is attached to the actual geometry classes.



6.4.2 Metadata on Attribute Level

Additionally, metadata can be stored on attribute level (third level), for example to allow specifying that a particular attribute may be collected from another source than the feature itself. (It can be foreseen that the width of a **Bridge** is collected by an engineer.)

The metadata is associated to the **datatype** *TypeMeta* from which all **attributes** inherit within the triple construct on the second level (see 6.3.5)



6.5 Associations

Associations connect elements of the model describing relations between them. They should only be unidirectional. Only in exceptional cases may bidirectional associations be used.

In DGIM three different types of relations are used.

6.5.1 Associations between featureTypes

Associations between featureTypes represent a semantically relation between featureTypes that cannot be discovered by analysis of the model. Hence associations shall not be used to represent spatial relations (except beyond 2D representation).

Examples for allowed associations

- PopulatedPlace isCapitalOf GeopoliticalEntity
- AirTrafficControlTower controls Airspace

Examples for not allowed associations (since those can be found by spatial analysis)

- City liesWithin AdministrativeDivision
- Threshold belongsTo Runway

6.5.2 Associations between featureTypes and types

Types usually describe additional information independent of spatial relations. They can be associated to featureTypes to provide detail information or specific domain related specification. Normally the featureType is associated with a unidirectional link to one or more types.

6.5.3 Associations between types

Associations between types should be avoided since this cause another level of complexity especially if by associating types chains are created. Most commercial software products will have difficulties to handle those. The restriction may be removed as developments progress.

7 Content

The content of the DGIM is managed by a maintenance team.

Although the maintenance team has responsibility for how the model is developed, the basic idea and some basic rules are pre-defined.

The model itself is meant to be primarily for the exchange of information. Therefore some aspects in the model are not optimized for collection of data or for representation in a web service.

7.1 Basic Principles

7.1.1 Generic Model

The model shall be generic enough to allow a general classification of a real world phenomenon. For example:

LandTransportationWay is used instead of Road, Drove, CartTrack, etc. since it describes a way without the need to narrow the meaning on feature level. Additional attribution allows for the description of the physical characteristics in more detail.

InlandWater is used instead of River, Pond, WaterRace, etc. for the same reason. It should be possible to describe the pure existence of inland water before any classification is done.

7.1.2 Physical Representation vs. Thematic Meaning

The model tries to separate the physical representation from the actual usage or meaning of the phenomenon. This results in two aspects:

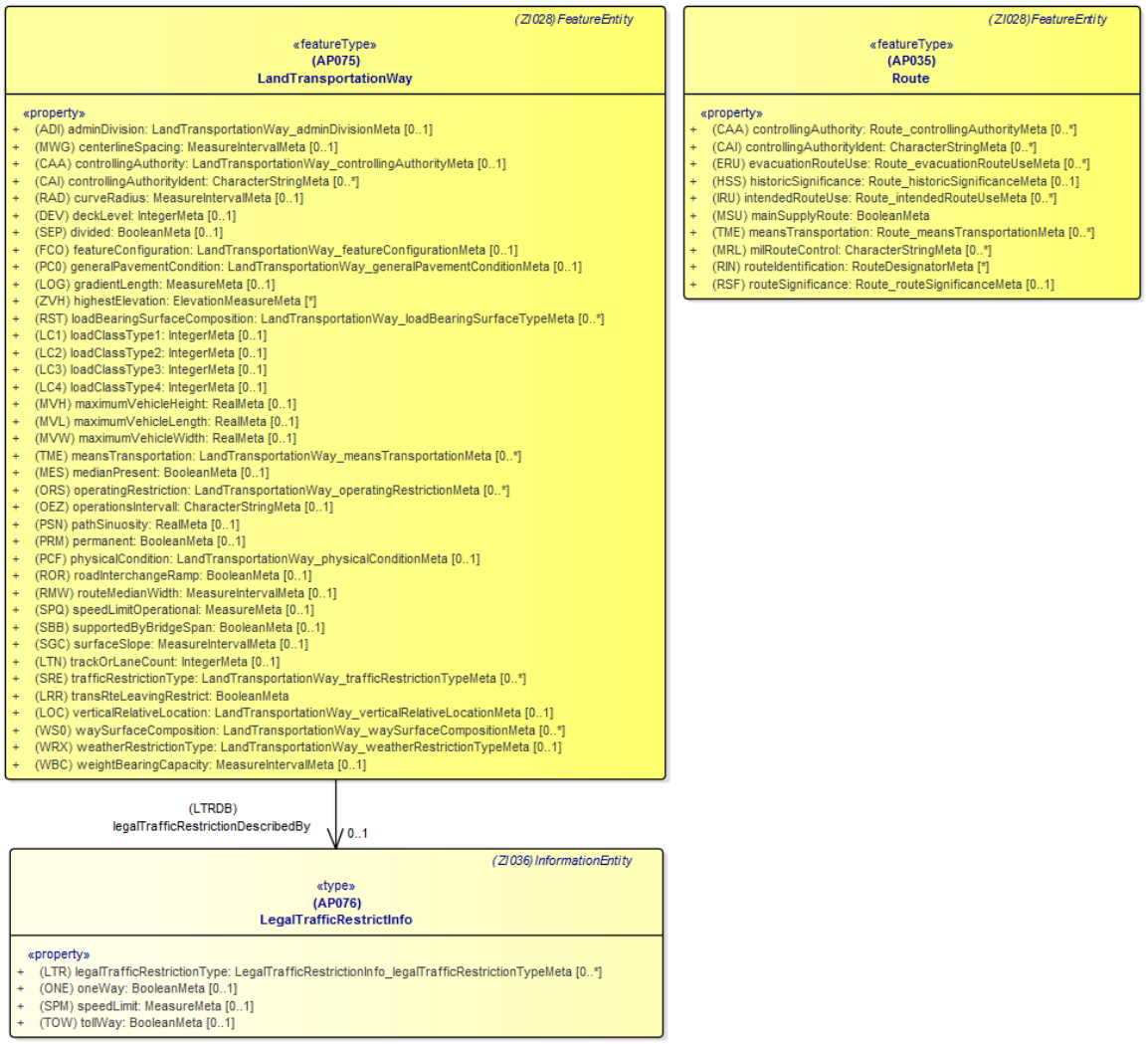
There could be a phenomenon that is defined/described by more than one feature to allow describing both aspects to be described independent from each other.
For example, a city modelled as BuiltUpArea describing the physical appearance (buildings, heights, material, etc.) and as PopulatedPlace describing the function and meaning (capital, trading centre and settlement type, etc.).

The second aspect shows the difference between “physical” featureTypes such as Building, River or Forest and the thematic featureTypes such as route, conservation area, exclusive economic zone, etc. that may be attached with physical featureTypes or maybe completely artificial without a representation in the real world.

7.2 Basic Model Constructs

This chapter describes general model principles regarding the content of the model.

7.2.1 Land Transportation Network



The principles stated in chapter 7.1 led to design decisions regarding the basic content of the model. In the case of the land transportation network, natural language often confuses with the pure facts in the real world. The term road may be quite clear to every human but the difference between a cart track, a road and a highway may be difficult though and even worse may be different from country to country. A highway in the USA or in Europe may be significantly different from a highway in central Africa. The idea in this model (since it is meant for exchange and not for direct user presentation or analysis) is to create a land transportation model that is as generic as possible without losing the "link" to spoke language that is hence represented by realWorldObjects.

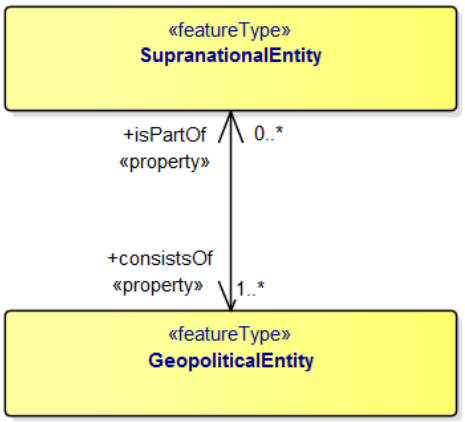
The current model consists of the feature LandTransportationWay, covering the physical representation of any kind of way from motorways to foot paths and Route to cover the more thematic driven network aspects. The information class LegalTrafficRestriction represents legal restrictions and many realWorldObjects providing the "translation" to terms such as highway, evacuation route, road, and hiking path.

7.2.2 Administrations

Administrations have been modelled to cover any kind of hierarchy/structure in any country/territory.

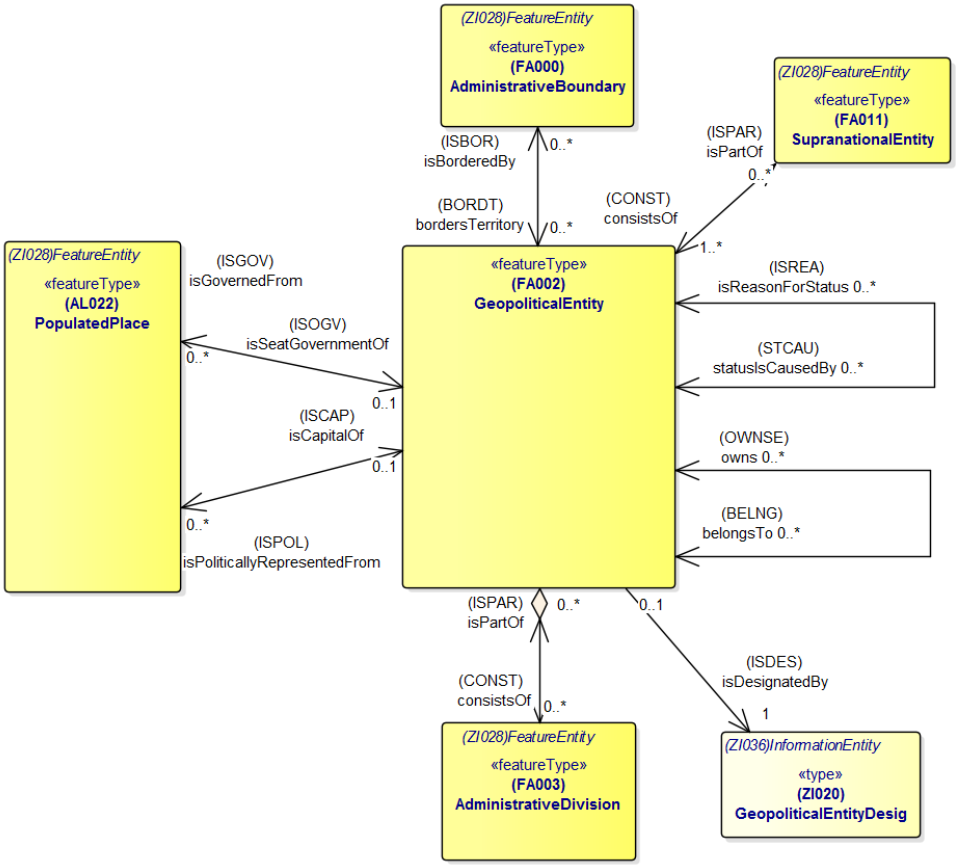
7.2.2.1 Supranational Entity

The highest level of administration unit is a *SupranationalEntity*, an organisation as the United Nations, the European Union or the North Atlantic Treaty Organisation.



7.2.2.2 Geopolitical Entity

A *GeopoliticalEntity* could be a formally acknowledged country but also territories as occupied areas, condominiums or similar. They do not belong to a hierarchy below another *GeopoliticalEntity* but could consist of *AdministrativeDivisions* as a country can.



A *GeopoliticalEntity* (country) may own another *GeopoliticalEntity* (territory), for example if it is leased etc. A *GeopoliticalEntity* can also be responsible for a current status (e.g. occupied) of another *GeopoliticalEntity*.

A *GeopoliticalEntity* is designated/named by a *GeopoliticalEntityDesign* and bordered by an *AdminitrativeBoundary*. It can be part of a *SupranationalOrganisation* such as NATO or EU.

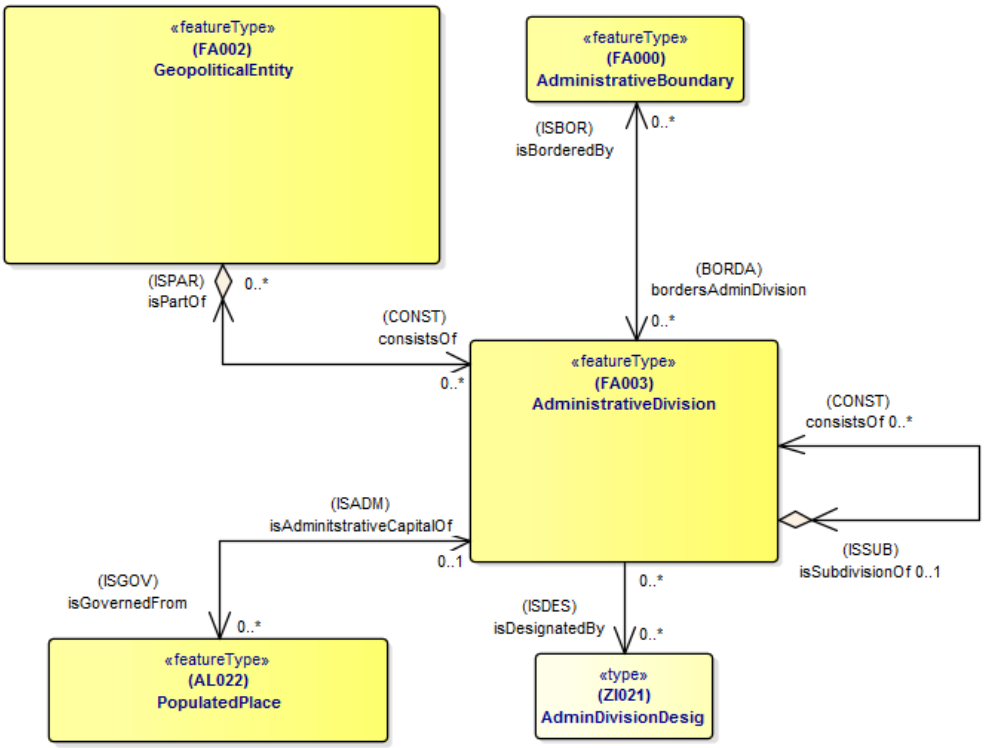
A *GeopoliticalEntity* may consist of A*dministrativeDivision*s, for example states in the USA or Bundeslaender in Germany.

A *GeopoliticalEntity* is governed from its seat of government that is a *PopulatedPlace* which can be different from the actual capital, the *PopulatedPlace* that is politically representing the *GeopoliticalEntity*, for example Amsterdam is the capital of The Netherlands while The Hague is the seat of government.

7.2.2.3 Administrative Divisions

An *AdministrativeDivision* is a political hierarchical organised entity within the *GeopoliticalEntity*. It starts with the highest level, which for example is a state in the USA or a Bundesland in Germany and goes down as required to municipal or even lower. The challenge in modelling administration is the diversity of these hierarchical systems that differ from country to country. To a most flexible and precise description of these systems, the *AdministrativeDivision* is classified in first place by two attributes:

The attribute *adminDivision* contains a number telling the user on which level within the specific *GeospatialEntity* this *AdministrativeDivision* is positioned. The attribute *adminUnitTypeName* is a textfield providing not the actual name of the *AdministrativeDivision* but the name of the level, for example "state" or "Bundesland".



7.2.3 Settlements

Settlements have been modelled guided by the principles stated above. Two **featureTypes** cover the main elements: A *PopulatedPlace* is the geographical, thematical, on a name based entity that covers a settlement. It focuses not on physical information but on thematic topics such as function. A *PopulatedPlace* may consist of one or more *BuiltUpAreas* that cover the physical representation of a settlement. *BuiltUpAreas* can cover a city, an industrial area or a trailer park.

PopulatedPlaces can therefore be connected to *GeospatialEntites* or *AdministrativeDivisions* to express their function as seat of government or capital.

Real World Objects such as shanty towns or permanent refugee camps are covered by these elements.

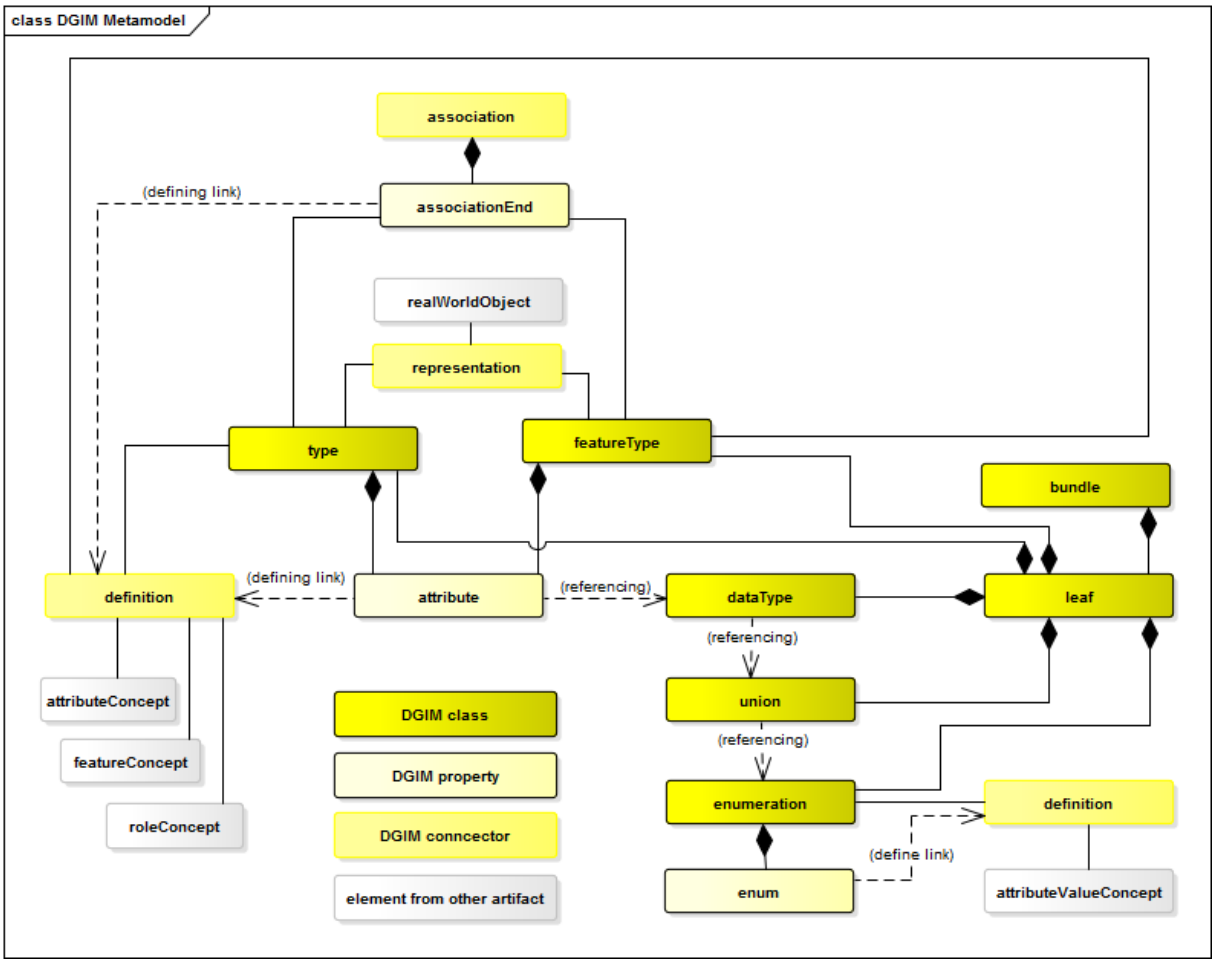
8 Conformance

Conformance to DGIF is achieved by being conformant to a DGIF product. Data Product Specifications define conformance classes.

9 Metamodel

9.1 Overview

The metamodel of DGIM mainly consists of **packages**[3], **entities/feature types**, **data types** of any kind and **enumerations**.



[3] A *package* is a means of structuring information in the modelling environment (DCE) that is used for maintaining the DGIF. It can be compared to a *folder* in a file manager.

9.2 Elements

9.2.1 Application Schema

Meta class: Package

The top **package** comprising the whole DGIM is of stereotype **application Schema**.

9.2.1.1 Properties

The information in Table 7 is attached to the **application Schema**.

Table 7: Information attached to application Schema

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
name	"DGIM"	String	1	YES		String
Stereotype properties (tagged values)						
gmlProfileSche-ma	In the case that the schema conforms to one or more GML Schema Profiles, their schema location(s). For example: http://schemas.opengis.net/gml/3.2.1/profiles/gmlSimpleFeatureProfile/1.1.0/gmlsf.xsd (defined by GML 3.2)	URI	0..1	NO		URI
targetNamespace	A URI for the unique XML namespace associated with this Schema. (defined by GML 3.2)	URI	1	YES		URI
version	The version number of this Schema. (defined by GML 3.2)	String	1	YES		String
xmlns	An abbreviation for the unique XML namespace associated with this Schema. (defined by GML 3.2)	String	1	YES		String
xsdDocument	The filename of the XML Schema Document (XSD) for this Schema. (defined by GML 3.2)	String	1	YES		String

9.2.1.2 Links

The **applicationSchema** consists of many **bundles** and **leaves**.

9.2.2 Bundle and Leaf

Meta Class: Package

A **package** bundles elements of a specific theme or from a specific source. In DGIM the packages can be **bundles** if they consist of other (sub-)**packages** or **leaves** if they just store elements that are not **packages**.

9.2.2.1 Properties

The information shown in Table 8 is attached to a **bundle** or **leaf**.

Table 8: Information attached to bundles and leafs

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(package) name	AlphaCode of the package	String	1	YES		String
Stereotype properties (tagged values)						
description	An optional statement of the nature, properties, scope, or non-essential qualities of the concept that are not specified by the definition.	String	0..1	NO		Memo
documentation	A precise statement of the nature, properties, scope, or essential qualities of the concept.	String	1	YES		String
name	Name of the category	String	1	YES		String
xsdDocument	XSD Document: The filename of the XML Schema Document (XSD) for this Schema. (defined by GML 3.2)	String	1	YES		String
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String

9.2.2.2 Links

The **bundles** and **leaves** are within the **applicationSchema** and consist of **elements** such as **classes** and **enumerations**. A **bundle** additionally can store many **leaves**.

9.2.3 FeatureType

Meta class: Class

The **featureType** represents an entity.

9.2.3.1 Properties

The information in Table 9 is attached to a **featureType**.

Table 9: Information attached to featureType

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(class) name	The alphaCode as defined in the DGFCD.	String	1	YES		String
alias	The 531-Code as defined in the DGFCD	String	1	NO	Derived from DGFCD	String
Stereotype properties (tagged values)						
byValueProper-tyType	A Boolean value where TRUE creates an additional property type that enforces that the Entity Type is encoded inline in the property in XML in-stances. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
isCollection	A Boolean value where if TRUE identifies that the Entity Type acts as an object collec-tion; the association role is converted to a property ele-ment. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
noPropertyType	A Boolean value where TRUE suppresses the creation of a property type in GML; usually set if the Entity Type is never a value of another property or only references are used to represent associations be-tween objects. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
geometry	The default allowed geome-tries for the featureType, e.g. "P,S" for point and surface. General order is "P,MP,C,MC,S,MS,So,MSo".	String	1	YES	default: " P,MP,C,MC,S ,MS,So,MSo"	String
profiles	Comma separated list of identifiers that define in which profiles the (feature) type is applicable. An additional parameter is "geometry" (ShapeChange extension)	String	0..1	YES		String
securityClassifi-cation	The security classification that applies to this entity.	Securi-tyClassifi-cation	1	YES		Securi-tyClassifica-tion

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
xmlSchemaType	If this Entity Type has a canonical XML Schema encoding (*e.g.*, from the XML Schema Definition language) then specifies the corresponding XML Schema *typename*. (defined by GML 3.2)	String	0..1	YES		String
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
status	The status of the item as defined in ISO 19135 ('valid', 'notValid', 'retired', 'superseded')	RE_ItemStatus	1	YES		RE_ItemStatus
dateAccepted	The date the item was accepted and became valid.	Date	1	YES		String
dateRetired	The date the item was retired.	Date	0..1	YES		String

9.2.3.2 Links

A **featureType** is linked to its own (and its **properties**) appropriate **concepts** in the DGFCD via a **connector** of type **definition**.A **featureType** is connected from **realWorldObjects**. The connection is of type **representation**.

A **featureType** has a **superclass** (either the class **featureEntity** or another **featureType**) and may have one or more **subclasses** connected by a **generalization**. These are also of type **featureType**.

A **featureType** may have **associations** to other **featureType**(s) or type(s) to express a semantic relation.

A **featureType** may have one or more **properties** to describe its characteristics.

9.2.4 Type

Meta class: Class

A **type** is a class representing information that is not of spatial nature. This can be an information class or a service.

9.2.4.1 Properties

The information shown in Table 10 is attached to a **type**.

Table 10: Information attached to type

Item	Definition/Content	Data type	Multiplicity	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(class) name	The alphaCode as defined in the DGFCD.	String	1	YES		String
Alias	The 531-Code as defined in the DGFCD	String	1	NO	Derived from DGFCD	String
Stereotype properties (tagged values)						
byValuePropertyType	A Boolean value where TRUE creates an additional property type that enforces that the Type is encoded inline in the property in XML instances. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
isCollection	A Boolean value where if TRUE identifies that the Type acts as an object collection; the association role is converted to a property element. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
noPropertyType	A Boolean value where TRUE suppresses the creation of a property type in GML; usually set if the Entity Type is never a value of another property or only references are used to represent associations between objects. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
profiles	Comma separated list of identifiers that define in which profiles the Type is applicable. (ShapeChange extension)	String	0..1	YES		String
securityClassification	The security classification that applies to this Type.	SecurityClassification	1	YES		SecurityClassification

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
xmlSchemaType	If this Type has a canonical XML Schema encoding (*e.g.*, from the XML Schema Definition language) then specifies the corresponding XML Schema *typename*. (defined by GML 3.2)	String	0..1	YES		String
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
gmlMixin	Allow for multiple inheritances when deriving schemas with ShapeChange	Boolean	1	YES		Boolean
status	The status of the item as defined in ISO 19135 ('valid', 'notValid', 'retired', 'superseded')	RE_ItemStatus	1	YES		RE_ItemStatus
dateAccepted	The date the item was accepted and became valid.	Date	1	YES		String
dateRetired	The date the item was retired.	Date	1	YES		String

9.2.4.2 Links

A **type** is linked to its own (and its **properties**) appropriated **concepts** in the DGFCD via a **connector** of type **definition**.A **type** is connected to **realWorldObjects**. The connection is of type **representation**.

A **type** has a **superclass** (either the class **actorEntity**, **eventEntity** or **informationEntity** or another **type**) and may have one or more **subclasses** connected by a **generalization**. These are also of type **type**. A **type** may have **associations** to other (**feature**) **type**(s) to express a semantic relation.

A **type** may have one or more **properties** (attributes) to describe its characteristics.

9.2.5 Data Type

Meta Class: UML Data Type

A **dataType** represents the actual nature of a **property**.

9.2.5.1 Properties

The information shown in Table 11 is attached to a **dataType**.

Table 11: Information attached to dataType

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(class) name	The alphaCode of the Data Type: A compact and human-readable designator that is used to denote the Data Type.	String	1	YES		String
Stereotype properties (tagged values)						
noPropertyType	A Boolean value where TRUE suppresses the creation of a property type in GML; usually set if the Data Type is never a value of another property or only references are used to represent associations between objects. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
xmlSchemaType	If this Data Type has a canonical XML Schema encoding (*e.g.*, from the XML Schema Definition language) then specifies the corresponding XML Schema *typename*. (defined by GML 3.2)	String	0..1	YES		String
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
description	A short description of the nature, usage and/or structure of the Data Type.	String	0..1	NO		String
status	The status of the item as defined in ISO 19135 ('valid', 'notValid', 'retired', 'superseded')	RE_ItemStatus	1	YES		RE_ItemStatus
dateAccepted	The date the item was accepted and became valid.	Date	1	YES		String
dateRetired	The date the item was retired.	Date	1	YES		String

9.2.5.2 Links

A **dataType** may have a **superclass** or one or more **subclasses** connected by a **generalization**. These are also of the same stereotype.

A **dataType** may have one or more **properties** (attributes) to describe its characteristics.

9.2.6 Union

Meta Class: UML Data Type

A **union** is a combination of different elements as a specific type.

9.2.6.1 Properties

The information in Table 12 is attached to a **union**.

Table 12: Information attached to union

Item	Definition/Content	Data type	Multiplicity	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(class) name	A compact and human-readable designator that is used to denote the Type	String	1	YES		String
Stereotype properties (tagged values)						
dateAccepted		Date	1	YES		String
dateRetired		Date	1	YES		String
status		RE_ItemStatus	1	YES		RE_ItemStatus
noPropertyType	A Boolean value where TRUE suppresses the creation of a property type in GML; usually set if the Union is never a value of another property or only references are used to represent associations between objects. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
xmlSchemaType	If this Union has a canonical XML Schema encoding (*e.g.*, from the XML Schema Definition language) then specifies the corresponding XML Schema *typename*. (defined by GML 3.2)	String	0..1	YES		String

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
description	A short description of the nature, usage and/or structure of the data type.	String	0..1	NO		String

9.2.6.2 Links

A **union** has at least two **properties** (attributes) representing the alternatives.

9.2.7 Property (Attribute)

Meta Class: UML Property

An **attribute** is a specific **property** in the sense of a feature catalogue.

9.2.7.1 Properties

The following information shown in Table 13 is attached to a **property**.

Table 13: Information attached to property

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(Property) name	The alphaCode as defined in the DGFCD.	String	1	YES		String
alias	The 531-Code as defined in the DGFCD	String	1	NO	Derived from DGFCD	String
Stereotype properties (tagged values)						
isMetadata	A Boolean value indicating that this Entity Attribute may be considered as metadata in accordance with the GML encoding rules. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
gmlImplement-edByNilReason	A Boolean value indicating that this Entity Attribute should be implemented using the GML *nillable* construct. (ShapeChange extension)	Boolean	1	YES	default: false	Boolean

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
inlineOrByRefer-ence	A value indicating how this Entity Attribute should be encoded in GML. One of: { 'inline', 'byReference', 'in-lineOrByReference' }. (defined by GML 3.2)	At-tributeEn-coding	1	YES	default: in-lineOrByRef-erence	AttributeEn-coding
maxOccurs	The maximum number of values for this property. Appli-cable in specific derived rep-resentations, for example flattened representations. (ShapeChange extension)	Integer	0..1	YES		Integer
profiles	Comma separated list of identifiers that define in which profiles the Property is appli-cable. A parameter max-Occurs allows further specifi-cations, e.g. "TM50(maxOccurs=2)" to overwrite maxOccurs for a specific profile. (ShapeChange extension)	String	0..1	YES		String
securityClassifi-cation	The security classification that applies to this Property.	Securi-tyClassifi-cation	1	YES		Securi-tyClassifica-tion
sequenceNumber	Unique index of this Entity Attribute within the Entity Type; may be used to tag derived products in order to ensure consistent Entity At-tribute ordering. (defined by GML 3.2)	Integer	1	YES		Integer
xsdEncodingRule	The rule to be followed when creating an XSD-based en-coding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
isFlatTarget	(only used for associa-tionEnds): This Boolean designates the element in a association that gains all information from the linked element in the flattening process.	Boolean	0..1	NO		Boolean

9.2.7.2 Links

A **property** belongs to a specific **element** (for example a **featureType** or **type**).

9.2.8 Enumeration

Meta Class: Enumeration

An **enumeration** represents the list of values for a **property** of a specific **dataType**. The values of an **enumeration** are called **enums**.

9.2.8.1 Properties

The information in Table 14 is attached to an **enumeration**.

Table 14: Information attached to enumeration

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(Enumeration) name	The combination of the alphaCodes of the Feature Type and the Feature Attribute as defined in the DGFCD separated by an Underscore ("_").	String	1	YES		String
Alias	The combination of the 531-Code of the Feature Type and the Feature Attribute as defined in the DGFCD separated by an Underscore ("_"). In the rare case that an enumeration is used by several classes the placesholder "ZZ000" is set for the class' 531-code.	String	1	NO		String
Stereotype properties (tagged values)						
dateAccepted		Date	1	YES		String
dateRetired		Date	1	YES		String
status		RE_ItemStatus	1	YES		RE_ItemStatus
profiles	Comma separated list of identifiers that define in which profiles the Enumeration is applicable. (ShapeChange extension)	String	0..1	YES		String
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String

9.2.8.2 Links

An **enumeration** consists of **enums** and is referenced by a **dataType**.

An **enumeration** is linked to the appropriate **enumValueConcepts** in the DGFCD via a connector of type **definition**

9.2.9 CodeList

Meta class: Data Type

A **codeList** is a kind of **enumeration** that is expandable and managed in an external online environment/resource.

NOTE: In the current version of DGIF **codeLists** are represented by other **dataTypes**.

9.2.9.1 Properties

The information in Table 15 is attached to a **codeList**.

Table 15: Information attached to codeList

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(CodeList) name	The designation of the codelist.	String	1	YES		String
Stereotype properties (tagged values)						
asDictionary	A Boolean value indicating that a gml:Dictionary shall be used to represent the codeList. (defined by GML 3.2 and 3.3)	Boolean	1	YES		Boolean
codeList	An URI identifying a code list resource that corresponds to this code list. (ShapeChange extension)	URI	1	YES		URI
xsdEncodingRule	The rule to be followed when creating an XSD-based encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
Description	A short description of the nature, usage and/or structure of the data type.	String	0..1	NO		String
Status	The status of the item as defined in ISO 19135 ('valid', 'notValid', 'retired', 'superseded')	RE_ItemStatus	1	YES		RE_ItemStatus
dateAccepted	The date the item was accepted and became valid.	Date	1	YES		String

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
dateRetired	The date the item was retired.	Date	1	YES		String

9.2.9.2 Links

A **codeList**Codelist is referenced by a **dataType**.

9.2.10 Enum

Meta Class: Property

An **enum** is a value within an **enumeration**.

9.2.10.1 Properties

The information shown in Table 16 is attached to an **enum**.

Table 16: Information attached to enum

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(Enum) name	The alphaCode as defined in the DGFCD.	String	1	YES		String
alias	The 531-Code as defined in the DGFCD	String	1	NO	Derived from DGFCD	String
Stereotype properties (tagged values)						
profiles	Comma separated list of identifiers that define in which profiles the Enum is applica-ble. (ShapeChange extension)	String	0..1	YES		String
xsdEncodingRule	The rule to be followed when creating an XSD-based en-coding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String

9.2.10.2 Links

An **enum** belongs to one specific **enumeration**.

9.2.11 Property (AssociationEnd)

Meta Class: UML AssociationEnd

An **asociationEnd** is a specific **property** that describes the role of an **element** (for example a **type** or **featureType**) in an **association**.

9.2.11.1 Properties

The following information in Table 17 is attached to a **property**. (They are similar to the property described in 9.2.7)

Table 17: Information attached to property

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
(Property) name	The alphaCode as defined in the DGFCD.	String	1	YES		String
alias	The 531-Code as defined in the DGFCD	String	1	NO	Derived from DGFCD	String
Stereotype properties (tagged values)						
isMetadata	(only used for attributes): A Boolean value indicating that this Entity Attribute may be considered as metadata in accordance with the GML encoding rules. (defined by GML 3.2)	Boolean	1	YES	default: false	Boolean
gmlImplement-edByNilReason	(only used for attributes): A Boolean value indicating that this Entity Attribute should be implemented using the GML *nillable* construct. (ShapeChange extension)	Boolean	1	YES	default: false	Boolean
inlineOrByRefer-ence	(only used for attributes): A value indicating how this Entity Attribute should be encoded in GML. One of: { 'inline', 'byReference', 'in-lineOrByReference' }. (defined by GML 3.2)	At-tributeEn-coding	1	YES	default: in-lineOrByRef-erence	AttributeEn-coding
maxOccurs	The maximum number of values for this property. Appli-cable in specific derived rep-resentations, for example flattened representations. (ShapeChange extension)	Integer	0..1	YES		Integer
profiles	Comma separated list of identifiers that define in which profiles the Property is appli-cable. (ShapeChange extension)	String	0..1	YES		String

Item	Definition/Content	Data type	Multi-plici-ty	Norm.	Generation	EA Datatype
securityClassifi-cation	The security classification that applies to this Property.	Securi-tyClassifi-cation	1	YES		Securi-tyClassifica-tion
sequenceNumber	Unique index of this Entity Attribute within the Entity Type; may be used to tag derived products in order to ensure consistent Entity At-tribute ordering. (defined by GML 3.2)	Integer	1	YES		Integer
xsdEncodingRule	(only used for attributes): The rule to be followed when creating an XSD-based en-coding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String
isFlatTarget	This Boolean designates the element in a association that gains all information from the linked element in the flattening process.	Boolean	0..1	NO		Boolean

9.2.11.2 Links

A **property** belongs to a specific **association**.

9.2.12 Association

Meta Class: association

An **association** links **elements** to express a semantic relation. Each end of the **association** is described by a **property** (**associationEnd**) of the linked **element** (Table 18).

Table 18: Linking with associations

Source Class	Source Role	Source Mult	Target Class		Target Role		Target Mult	Dir
Entity	Entity	1	Entity		Definition		1	
Stereotype properties (tagged values)								
profiles	Comma separated list of identifiers that define in which profiles the Property is applicable. (ShapeChange extension)		String	0..1	YES		String	
securityClassification	The security classification that applies to this Property.		SecurityClassification	1	YES		SecurityClassification	