"*Delivering Military Advantage through multi-national geospatial interoperability*"

DGIWG 205
Defence Geospatial Information Model (DGIM)

Edition 3.0

Dated: 19 July 2024

Document type:	Standard
Document date:	19 July 2024
Edition:	3.0
Superseded	DGIWG 205 Defence Geospatial Information Model Edition 2.0
Responsible Party:	Defence Geospatial Information Working Group (DGIWG)
Audience:	DGIWG participants and defence associates and liaisons. 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:	

Table of Contents

List of Tables

List of Figures

i. Executive Summary

The Defence Geospatial Information Model (DGIM) standard is an artefact of the Defence Geospatial Information Framework (DGIF) suite of standards.

The DGIM provides the logical data model for defence geospatial information, specifying the syntactic structure for the semantic content of the DGIF. DGIM establishes a clear, complete, and internally-consistent logical data model providing as basis for system-specific implementations and products.

This standard defines the DGIM conceptual and logical metamodel, explaining model concepts and elements in detail. It further describes both general and specific model patterns and principles, to include features, data types, metadata, and associations. Further information concerning the metamodel and its specific elements as represented in UML, is also detailed in the final section of this standard.

DGIM, as a component of the DGIF (DGIWG 200), is desigend to be utilized with DGIWG 206 (Defence Geospatial Feature Concept Dictionary) and DGIWG 207 (Defence Geospatial Real World Object Index). A DGIF GML Encoding specification (DGIWG 208) is also available.

ii. Submitting organizations

Nation	Organization
Spain	Army Geographic Center

iii. Document point of contact

All questions regarding this document shall be directed to the editor (secretariat@dgiwg.org) or the contributor organisations:

iv. Revision history

Date	Release	Primary clauses modified	Description
2024-07-19	3.0	All	Publication Draft

v. Future work

Document will be revised as set out in the DGIWG Program of Work (PoW).

1 Introduction

The Defence Geospatial Information Model (DGIM) standard (DGIWG 205) is an artefact of the Defence Geospatial Information Framework (DGIF) suite of standards.

The DGIM is a key component of the DGIF providing the logical data model for defence geospatial information. It specifies the syntactic structure for the semantic content specified in the Defence Geospatial Feature Concept Dictionary (DGFCD) standard (DGIWG 206). DGIM establishes a clear, complete, and internally-consistent logical data model for defence geospatial information, serving as the basis for system-specific implementation models, schemas, and products.

The Defence Geospatial Real World Object Index (DGRWI) standard (DGIWG 207) is a separate DGIF artefact that can serves as an alternate entry point into the DGIM. The DGRWI specifies real-world geospatial phenomena and links them to their model representation in the DGIM.

2 Scope

The DGIM specifies a technology neutral logical data model for the defence geospatial community. As a Platform Independent Model (PIM), DGIM determines the syntactic structure of the semantics used in the DGIF. Using Model Driven Architecture (MDA) techniques and technology-tied Platform Specific Models (PSM), physical data models may be automatically derived from DGIM 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 modeling.

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 DGIF compliant systems. The DGIM also provides the foundation for development of DGIF-based Data Product Specifications. The DGIM reduces the cost of evolving system-specific implementations to meet evolving systems, missions, and customer requirements while promoting data agility for DGIF users.

The DGIM leverages and integrates geospatial information requirements and modeling practices from multiple communities (MGCP, IPHG, IHO, NATO JGSWG, etc.) and established community data models (MGCP, AIXM, S-100, GGDM, NAS, etc.) whose data can be ingested and exchanged by DGIF compliant systems.

The DGIM is managed and maintained in UML. An important geospatial modeling element of the DGIM is the featureType. 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 modeling 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 modeling entities that may repre-

sent other geospatial information that does not necessarily correspond to a “real-world phenomena.” (such as elements of human geography). Unless otherwise stated, the terms feature and (modeling) entity are used interchangeably in this standard.

Individual items of feature and/or attribute information 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, NATO, etc.) and civilian sectors (e.g., IHO, ICAO/EUROCONTROL, WMO).

Where possible, information traceability is established from concepts in the DGIM to their specification in the DGFCD, and to appropriate authoritative concept sources 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?*”

3 Conformance

DGIM does not specify specific conformance clauses or tools. Conformance to the DGIF (DGIM) is achieved through conformance to a DGIF-based product specification. Individual Data Product Specifications define conformance classes.

4 References

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 modeling.

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.

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 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*
DGIWG 200 – *Defence Geospatial Information Framework (DGIF) 3.0*
DGIWG 206 – *Defence Geospatial Feature Concept Dictionary (DGFCD) 3.0*
DGIWG 207 – *Defence Geospatial Real World Object Index (DGRWI) 3.0*
DGWIG 208 – *Defence Geospatial Information Framework Encoding Specification: (GML) 3.0*

The informative (non-normative) documents listed in Table 2 are useful to understanding and using this standard. For dated references, only the cited edition or version applies.

Table 2: Informative References

Standard or Specification
ISO 19111:2003 - *Geographic information – Spatial referencing by coordinates*
ISO 19112:2003 - *Geographic information – Spatial referencing by geographic identifiers*
ISO 19115-1:2014 - *Geographic information – Metadata - Part 1: Fundamentals*
ISO 19123:2005 - *Geographic information – Schema for coverage geometry and functions*
ISO 19136:2007 - *Geographic information – Geography Markup Language (GML)*
ISO/IEC 19507:2012 - *Information technology -- Object Management Group Object Constraint Language (OCL)*
DGIWG 114 - *DGIWG Metadata Foundation 2.0.0*

5 Terms, definitions, and abbreviations

5.1 Definitions

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

Table 3: Definitions applicable to this standard

Term	Definition
Entity	A modeling class that may represent either a feature or other geospatially-located information.
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.

5.2 Abbreviations & Acronyms

The acronyms used in this standard are specified Table 4.

Table 4: Acronyms Applicable to this Standard

Acronym	Definition
AIXM	Aeronautical Information Exchange Model
DCE	DGIF Collaborative Environment
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
DTM50	Defence Topographic Map, 1:50,000 Scale
EUROCONTROL	European Organisation for the Safety of Air Navigation
GGDM	Groundwarfighter Geospatial Data Model
GML	Geography Markup Language
ICAO	International Civil Aviation Organization
IEC	International Electrotechnical Commission
IHO	International Hydrographic Organization
IPHG	International Program for Human Geography
ISO	International Organization for Standardization
JGSWG	Joint Geospatial Standards Working Group
MDA	Model Driven Architecture
MGCP	Multinational Geospatial Coproduction Program
NATO	North Atlantic Treaty Organization
NSG	National System for Geospatial-Intelligence
ONINA	Other No Information Not Applicable
OCL	Object Constraint Language
OMG	Object Management Group
PIM	Platform Independent Model
PSM	Platform Specific Model
SI	International System of Units
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
UUID	Universally Unique Identifier
VMST	Vector Models and Schema Team
WMO	World Meteorological Organization
XML	Extensible Markup Language
XSD	XML Schema Document

6 Logical Structure

6.1 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 outlined in Figure 1 and described in greater detail in Section 9. The ISO 19109 metamodel has also been partly simplified and additional classes are added to support the organization and presentation of DGIM content.

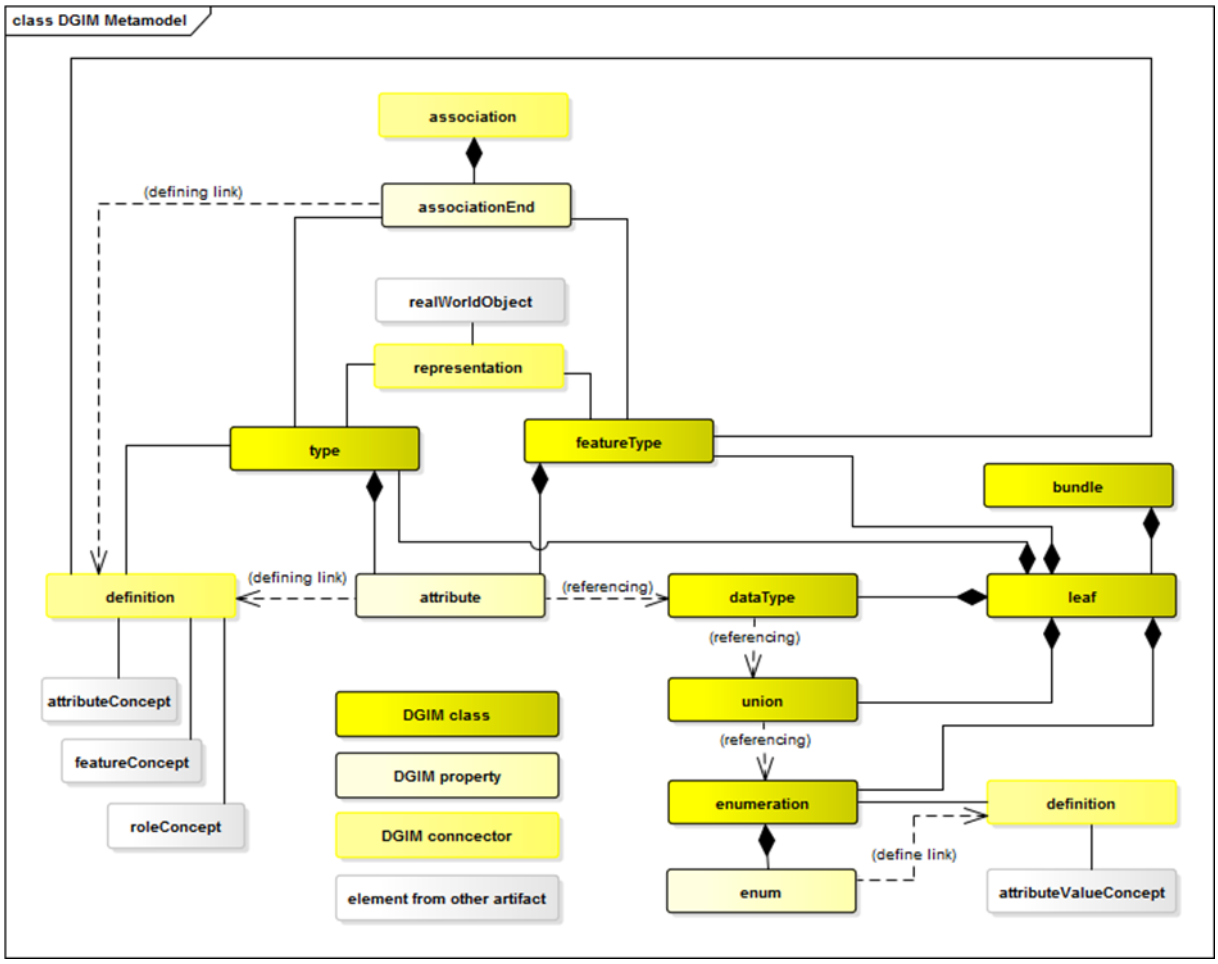

Figure 1: DGIM Metamodel, conformant to the ISO 19109 General Feature Model.

As necessary, the metamodel specified in ISO 19109 has been extended based on the object modeling component of ISO/IEC 19505-2:2012 - *OMG Unified Modeling Language UML, Superstructure*. In particular, the class and datatype metamodels from the Kernel package have been used for extension guidance.

6.2 Logical Metamodel

The DGIM metamodel is described below. The subsequent sections of this standard describe model elements and components using "camel case." Camel case is used in UML modeling and is a case-sensitive naming convention that differentiates modeling elements through the use of word capitalization and elimination of spaces between words or abbreviations. Camel

case is used extensively in this document (e.g. “featureType,” “dataType,” “alphaCode,” etc.) Where possible, descriptions and discussions of elements retain their camel case special capitalization and space elimination in the text of this standard.

The DGIM metamodel is organized into five basic types of Elements which are shown in Figure 1 and defined in Section 6.3. These are:

1. Feature Type (featureType)
2. Type (type)
3. Data Types (dataType)
4. Union (union)
5. Enumerations (enumeration)

The Model includes additional “Bundle” and “Leaf” elements supporting the management of the DGIM (see section 6.2.3 and 6.2.4 for more information).

Feature Types (“featureTypes”) and “types” are characterized by properties called “attributes” and “roles.” “Enumerations (enumeration)” contain “enums” as valid values for certain attributes.

A type and featureTypes can be linked using “associations” with specific properties called “association roles.”

Generalisations between types or featureTypes can be used to inherit information and to build hierarchical dependencies.

The DGIM Elements are linked to the Concepts defined in the DGFCD (“Feature Concepts,” “Attribute Concepts,” “Role Concepts” and “Attribute Value Concepts”) to gain precise meaning using the “Definition” element.

Combinations of elements defining a specific meaning can be designated by defining “Real World Objects” in the DGRWI using a “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.

6.2.1 General Model Specifications

Each model element in DGIM requires specific information as following:

- **(Class) Name:** The class name represents the primary 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. It is also called a primary code and represents a human readable name that can be longer than 25 characters. It should be used as primary identification. In cases where an information system may be unable to use identifiers of this length, an Alias may be used.
- **Alias:** The alias represents the 531Code, a unique alphanumeric value defined in the DGFCD (DGIWG 207 ed. 3.0) 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 where the alphaCode cannot be used in the system environment.

- **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.
- **DateAccepted:** The date when the element was accepted to be a valid element with the DGIM.
- **DateRetired:** The date when the element was retired from the DGIM and became *retired* or *superseded*.
- **SecurityClassification:** The specific security specification of the element. This concept is not implemented in DGIM 3.0.

6.2.2 Multiplicity

The information defining whether a property is mandatory or optional is documented by the multiplicity as shown in Table 5:

Table 5: 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 values 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.

Additional information may be 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: GML.

6.2.3 Bundle

A bundle[1] is 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.

6.2.4 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.

Leaves are created by themes based on the content.

[1] Occasionally, featureTypes may reside in the bundle hierarchy level if the creation of a theme is necessary (e.g. because there is only one element).

6.3 Model Elements

6.3.1 FeatureType (featureType)

A featureType is 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.

taggedValues give more information about a featureType. The taggedValue geometry specifies the different geometries allowed for the featureType (See Section 7.2). 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.

featureTypes are connected to DGFCD featureConcepts by “definition” that links a given featureType to its name, definition and description maintained in the DGFCD. Similarly, featureTypes are connected by “definition” to all attributeConcepts and all roleConcepts defining attributes or roles used by the featureTypes. A featureType is also connected to a realWorldObject (DGRWI) for which the concept is a representation.

Figure 2 depicts the example featureType Market with its properties defined by the DGFCD and as a subtype of featureType Feature Entity.

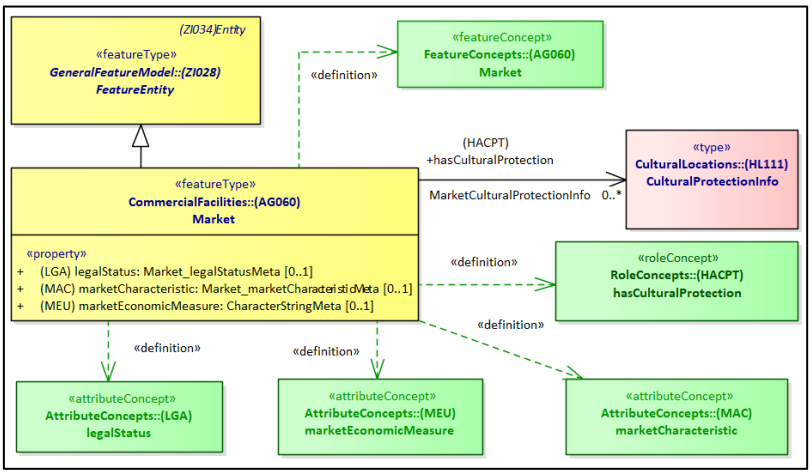

Figure 2: Excerpt from DGIM UML Model depicting a Market as a subtype of FeatureEntity and with attributes defined by the DGFCD.

6.3.2 Type (type)

A type is an information entity that has no inherent geometry. For example, *ReligiousInformation* that specifies religious significance and/or characteristics but is a phenomenon that has no position or spatial extent. In the context of the model, a type behaves as featureType without geometry.

6.3.3 Data Type (dataType)

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.

A dataType is used to construct the data type triple (see section 7.3.5) and to establish complex types that consist of several (sub)properties. For example, *HeightMeasure* (see Figure 3), comprises not only of the actual value but also some properties to store accuracy. To explain the nature of the dataType a description can be populated.

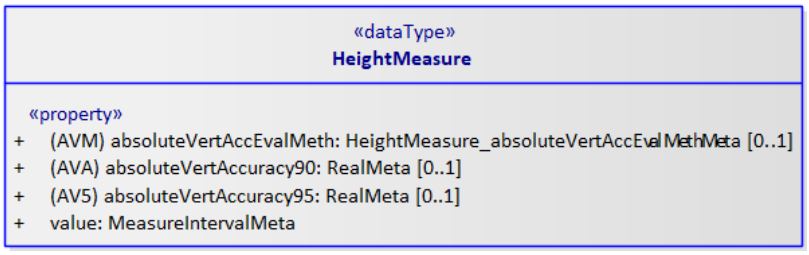

Figure 3: Example datatype HeightMeasure.

6.3.4 Union

A union is a special datatype. All properties stored in a union can be understood as alternatives. For example, the union *MeasureReason* (see Figure 4) represents a measured value or an ONINA (Other, No Information, Not Applicable) value.

Figure 4: Example union for MeasureReason.

6.3.5 Attribute

Attributes are properties of elements describing specific characteristics. Attributes have a specific dataType. They have a “multiplicity” that defines how often an attribute appears or how many values are allowed for an attribute (see Figure 5). A multiplicity starting with “1” defines a mandatory property for the element. Multiplicities starting with “0” define an optional property for the element.

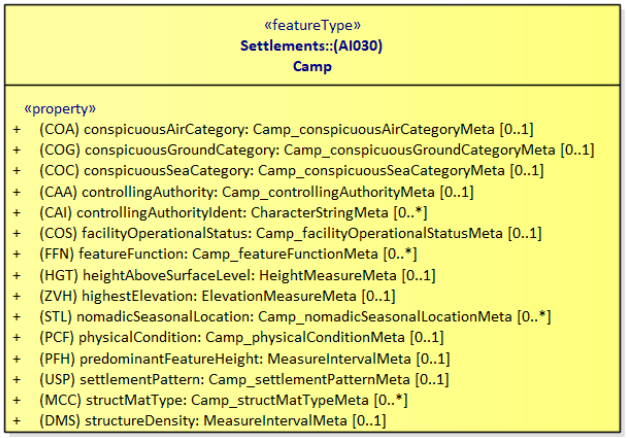

Figure 5: Example of feature attributes with indefinite (0..*) and optional, exactly one (0..1) multiplicity.

For an indefinite multiplicity (*) the taggedValue maxOccurs limits the actual number of values[2]. 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.

6.3.6 Associations

Associations specify a semantic or structural relationship between elements. “Composition” and “Aggregation” relationships, as defined by UML, are additional types of associations within this model. Associations are usually used to link information (types) to features (featureTypes) (see Figure 6 for an example). Associations are accompanied by “associationRoles” which contain further information about the type of specified relationship.

Associations from featureType to featureType are generally avoided in DGIM. In cases where multiple featureTypes are strongly associated, but not spatially coincident, an association may be used to specify a feature to feature relationship; however, the use of featureType to featureType associations must be carefully considered. Extensive use of associations introduces significant complexity to the process of creating, managing, and implementing schema derivations. DGIM assumes the use of spatial analysis tools within a DGIM-based physical implementation to identify spatial relationships between spatially-coincident featureTypes. More information on the usage of associations is given in Section 7.5.

[2] A default value of 3 is used in the derivation of schemas created using ShapeChange software even if the **maxOccurs** value is empty.

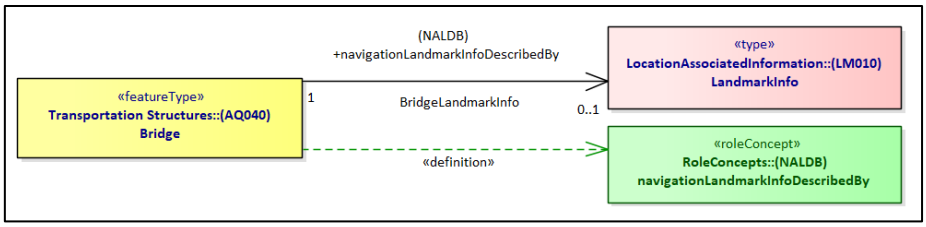

Figure 6: Example of a Describedby Association and associationRole.

6.3.7 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.

6.3.8 Enumeration

An enumeration represents the range of valid values for a given property. Sometimes called "pick list," only one or more of the values (enums) within this list is an allowed value for the property (see figure 7).

Allowable enumerations can be restricted as part of a DGIM profile. The taggedValues profiles are linked to all attributeValueConcepts that define the enums in the enumeration via definition.

Figure 7: Example of valid enumerations for an attribute.

6.3.9 Enum

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

6.3.10 Codelist

Codelists are external maintained and referenced lists of values. As of DGIF 3.0, Codelists are not utilized in the DGIM.

7 Modeling Patterns and Principles

7.1 General Feature Model

The DGIM is based on three different (abstract) entities inherited from the root class Entity (see Figure 8). These are: FeatureEntity, EventEntity, , and ActorEntity.

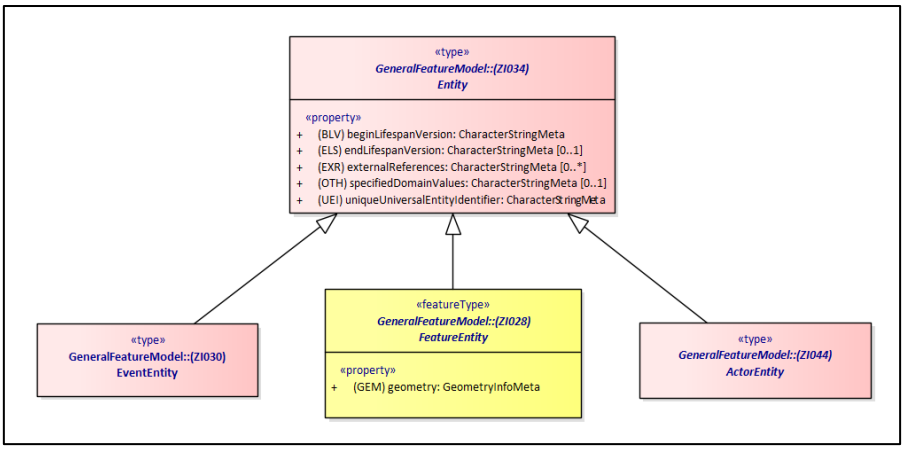

Figure 8: DGIM abstract entities.

Each abstract Entity has five common attributes:

beginLifespanVersion: Date and time at which this version of the spatial object was inserted or changed in the spatial data set.

endLifespanVersion: Date and time at which this version of the spatial object was superseded or retired in the spatial data set.

externalReferences: A list of external references that may provide additional or ancillary information about this entity instance.

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 (The actual attribute domain values may have been previously, or may become in the future, valid members of the attribute domain range).

uniqueUniversalEntityIdentifier: A globally exclusive, distinct, and persistent identifier assigned to an entity (ex. feature, event, actor, etc.). This identifier is used to support such things as: the unique identification of any individual instance of an entity, relationships between features in a relational database, and version control.

7.1.1 Feature Entity

An Entity that describes a feature defined by ISO 19110 as a representation of temporally persistent real-world phenomena, including their geometric position and extent. For example *Bridge*, *Building*, or *Harbour*. All featureTypes inherit from featureEntity.

featureEntity owns a single attribute:

The attribute *geometry* is of type *GeometryInfoMeta*. This attribute provides the creation of geometry for a featureType. The *GeometryInfoMeta* class is described in section 7.1.3.

7.1.2 Event Entity

An Event Entity is a temporal Entity describing the interaction of physical entities at a specific time or over a period of time that may or may not be have position and extent. In DGIM 3.0, *Events* cover a variety of types, which are further specified via attributes and associations.

7.1.3 Actor Entity

An Actor Entity is an abstract modeling entity that is a superclass for actor types. Actor types are representations of intentional entities (i.e. purposeful) that act or have the capability of acting as a participant in an event.

An actor may also own or control resources and/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 and an area of operation or action.

7.2 Geometry Model

The DGIF geometry model is based on ISO 19107. The following geometries are allowed in DGIF: *point*, *multipoint*, *curve*, *multicurve*, *surface*, *multisurface*, *solid* or *multisolid*. A detailed description of these geometries can be found in ISO 19107.

In addition to its inherent nature, *geometry* also owns an attribute linking ISO19107 elements to corresponding geometric attributes. Every *point* owns length, width, area and base elevation. In addition:

- Every *curve* owns width, area and base elevation since. Length can be derived from the geometry itself.
- Every *area* owns base elevation.
- Multigeometries (*multicurve, multisolid, multisurface*) do not own collective geometric attributes as each component geometry owns specific geometric attributes and represent include multiple different entities.

Attributes enable the storage of geometric information that cannot be measured or directly derived from the geometry itself. For example, legal (or official) runway length which may be shorter than the physical geometric extent.

Any featureType in DGIM can be represented by any geometry. The *geometry* attribute is inherited from *featureEntity* with an indefinite multiplicity (*). Possible geometries can be restricted in the derivation process for product-specific models and profiles. The taggedValue *geometry* restricts the possible geometries. For example, *BuiltUpArea* can have *point*, *curve* or *surface* geometry, but may be restricted to *point* for a small-scale profile or product (see Figure 9). Additionally, Metadata describing horizontal accuracy and vertical accuracy is associated with *GeometryInfo* (See Figure 10).

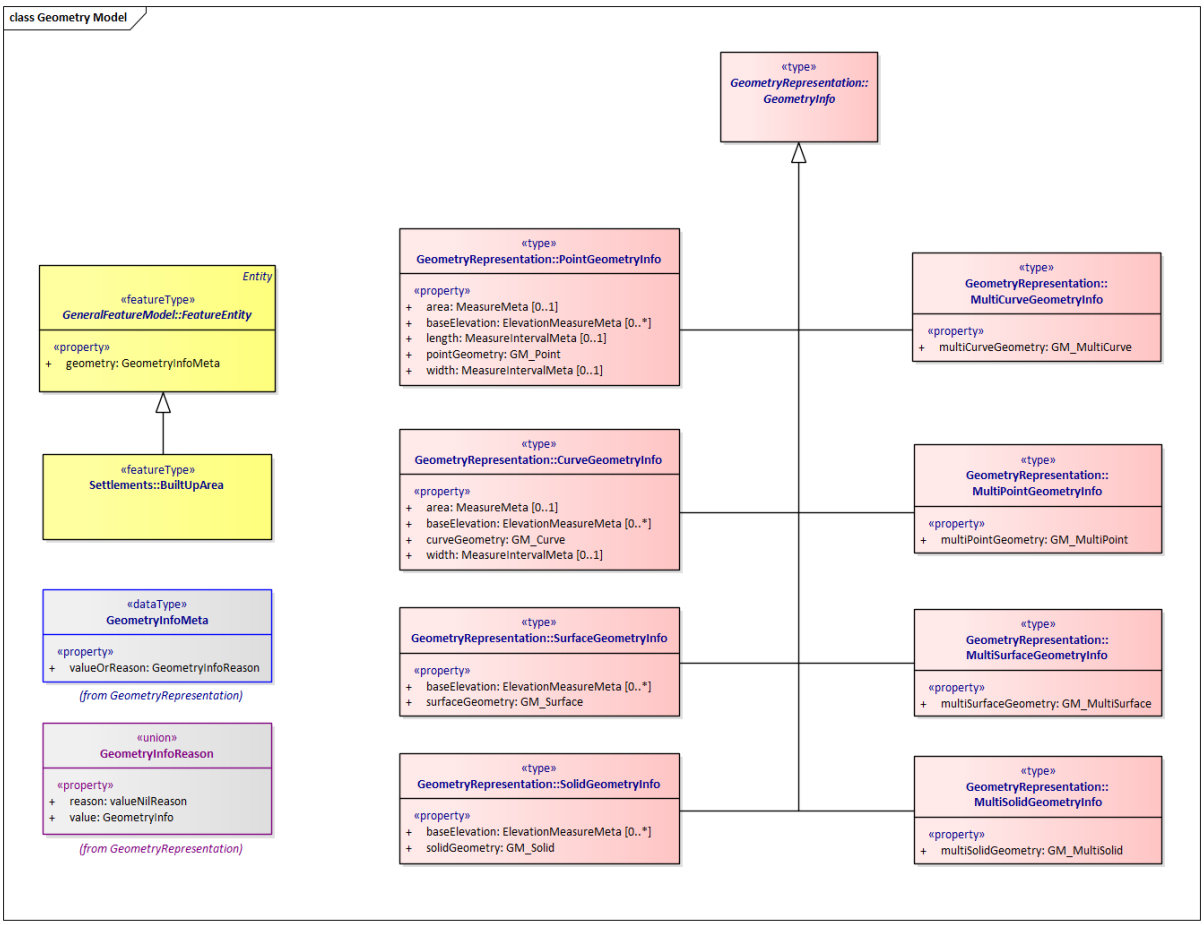

Figure 9: Geometry elements of the DGIM.

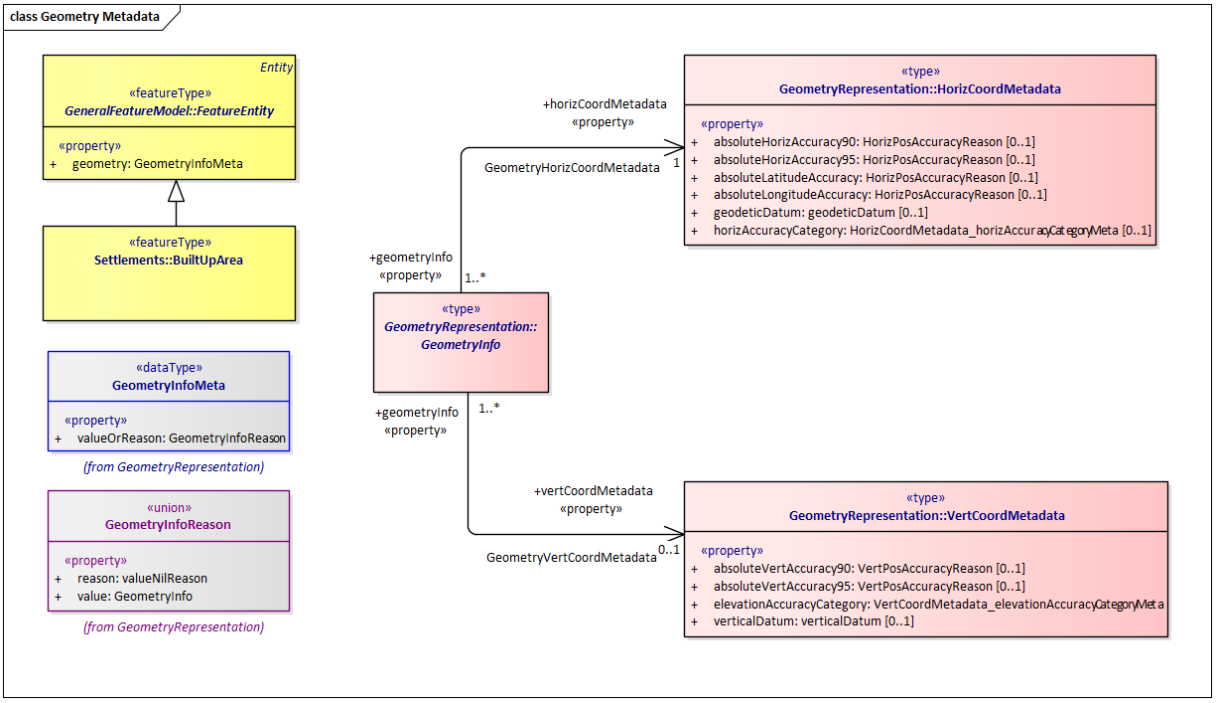

Figure 10: Example of restricted geometry elements and associated geometry metadata.

7.3 Modeling Data types (dataTypes)

7.3.1 Basic

Basic dataTypes are derived primarily from ISO 19103 and represent single values like text or a number. Six data types are used in DGIM 3.0:

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’

7.3.2 Complex

Complex dataTypes represent values accompanied by additional information such as accuracy, referenceSystem or other specialized information. Complex dataTypes are created based on the nature of the information being captured.

For example, the dataType *ElevationMeasure* is a complex dataType that consists of the *value* but also includes a property for *accuracy* and a property for *verticalDatum* (see figure 11).

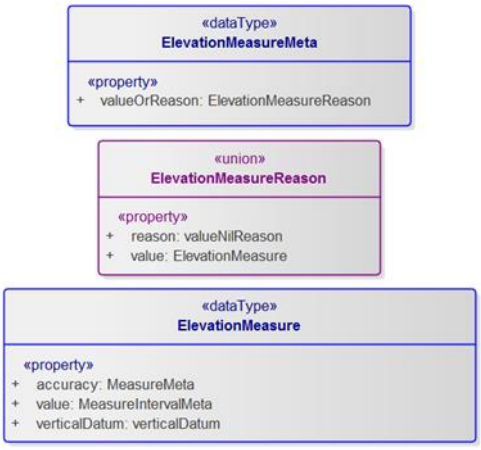

Figure 11: Example Complex dataTypes.

7.3.3 Enumerations

Enumerations specify a controlled list of allowed values (e.g. “blue, red, yellow, white” or “left, right”) usually including all possible valid values for a property. In DGIF 3.0, specific enumeration lists are created for each featureType and/or attribute relation that utilizes enumerations.

7.3.4 Relations to DGFCD attribute dataTypes

In DGFCD, dataTypes are defined differently because the dictionary level is at a higher level of abstraction. Table 6 shows the relations that must be obeyed.

Table 6: Relations of attribute dataTypes 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

7.3.5 Triple

A range of attributes can be represented by dataTypes. Most dataTypes are single types like characterString, real, measure or similar. DGIF also includes a dataType Triples, complex structures of dataTypes and unions enabling attribute metadata and default nil reasons to be modeled. Nil reasons are called ONINA collectively and represent the absence of values for an attribute.

7.3.5.1 Level Hierarchy

A dataType Triple consists of three levels of dataTypes. These consist of “Meta,” “Reason,” and “Value” and are explained in greater detail in below. Table 7 provides an example of level hierarchy applied to ONINA.

Table 7: Level hierarchy of the ONINAS dataTypes

3 levels of data types				
Type of values	**Meta:** Including Metadata	**Reason:** Including ONINA values	**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	MeasureIntervalReason	MeasureIntervalUnion and MeasureInterval	Measure (ISO 19103)
Interval of Numbers	IntegerIntervalMeta	IntegerIntervalReason	IntegerIntervalUnion and IntegerInterval	Integer (ISO 19103)
Text	CharacterStringMeta	CharacterStringReason		CharacterString (ISO 19103)
Boolean	BooleanMeta	BooleanReason		Boolean (ISO 19103)

Level 1 (Meta): Each attribute’s type is a dataType with a suffix “Meta.” These dataTypes enable metadata for specific attributes. The level 1 structure includes an inheritance from the

class *typeMeta* which is connected to attribute metadata (*FeatureAttMetadata*) (See Figure 12).

CharacterStringMeta or *IntegerMeta* are examples of Level 1 hierarchy in DGIM 3.0. Among other capabilities, this model pattern enables the use of a dataQualityStatement for an attribute that may differ from the metadata of the overall featureType (e.g. if it was derived from a different source).

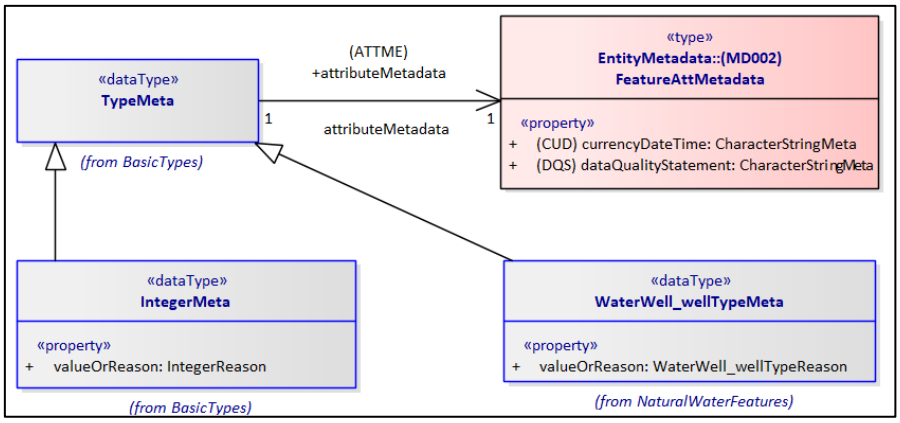

Figure 12: Level 1 dataType Hierarchy.

Level 2 (Reason): The second level is established by a union data type with a suffix “Reason” (e.g. “IntegerReason”). In a DGIF-based physical model (i.e. implementation), the union type allows the entry of an actual value (where it exists) or a standardized ONINA value (where an actual value does not exist) to express the null reason for a given attribute.

Level 3 (Value): The third level is the actual “Value” (see Figure 13). For simple types this is usually an integer or measure type derived from ISO19103. In the case of an enumeration dataType this is the actual enumeration specified in the DGIM.

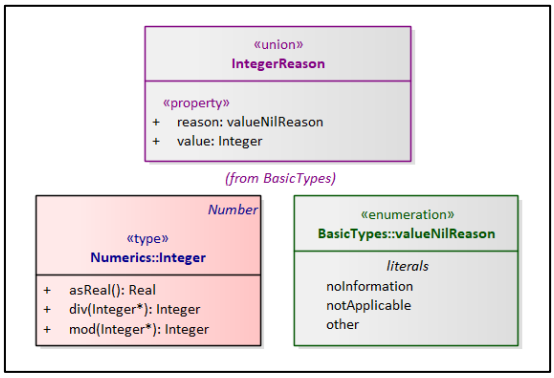

Figure 13: Example Actual Values.

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

For all enumeration types the whole triple has to be created individually and is stored in the thematic leaves (See Section 6.2.4) of the DGIM.

7.3.5.2 Triple Examples

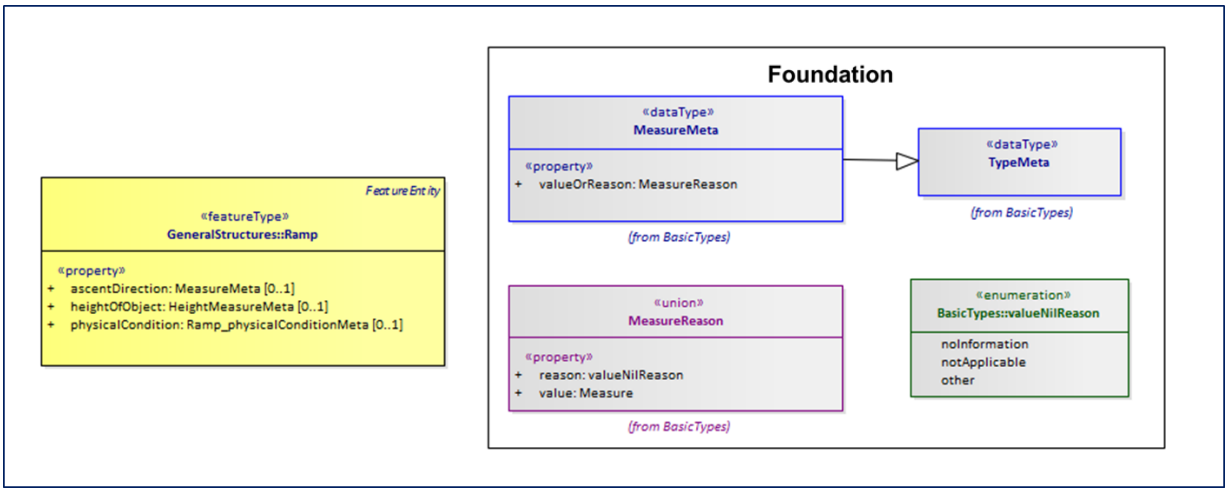

Figure 14: UML Example of dataType Triples using a general data type and ONINA values.

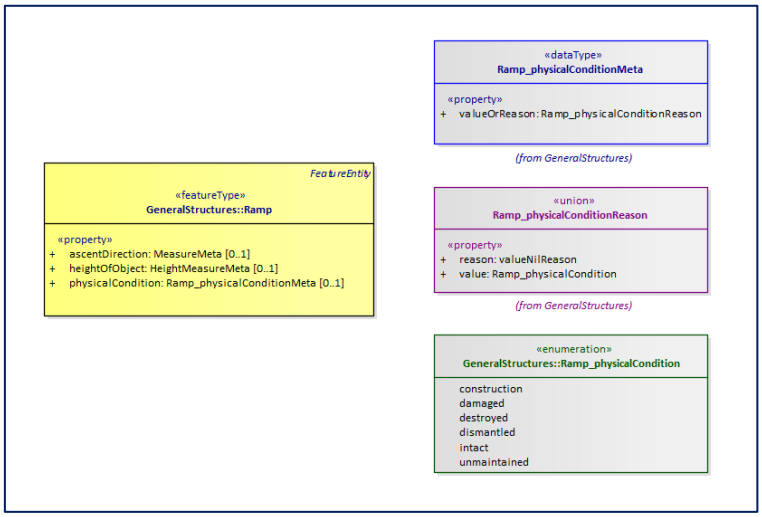

Figure 15: Example of dataType Triples depicting the use of actual values and ONINA values.

7.4 Metadata

Metadata for geospatial vector data is generally captured for data sets, features, and for attributes. Dataset metadata is not addressed within the DGIM specification. Dataset metadata, including DGIF-based datasets should utilize DGIWG Standard 114 - DGIWG Metadata Foundation (DMF) for dataset-level metadata.

7.4.1 Feature Level Metadata

Feature level metadata is described by *FeatureMetadata, SourceInfo and RestrictionInfo* (see Figure 16). Accuracy metadata is attached to the *geometry* classes as described in section 7.2.

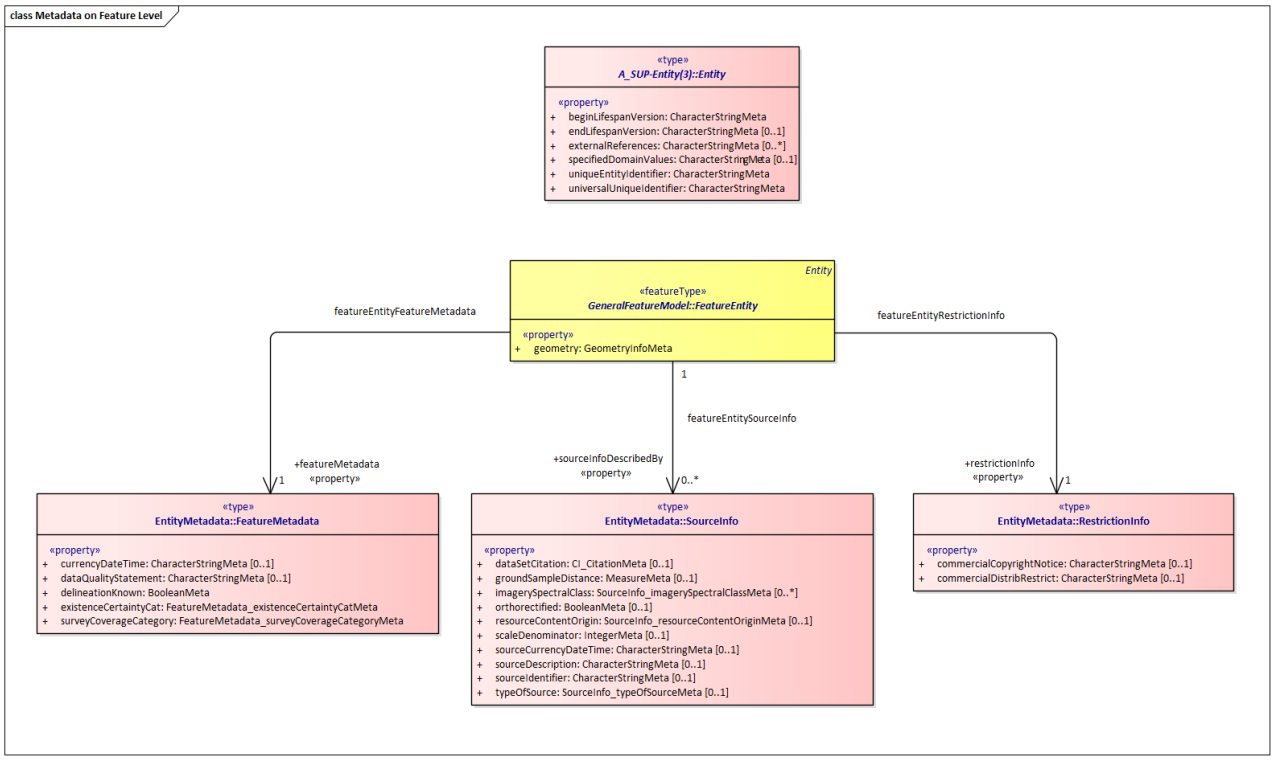

Figure 16: UML Example of Data Type Triples depicting the use of actual values and ONINA values.

7.4.2 Attribute Level Metadata

Attribute level metadata enables the specification of detailed and nuanced source information for particular attributes, especially in cases where a specific attribute value has been collected from another source. For example, the height value of an “Aerial” may have been collected by an engineer or provided by an ancillary source other than what was used to collect the feature (see Figure 17).

Attribute metadata is associated to dataType *TypeMeta* from which all attributes inherit within the second level triple construct discussed in section 7.3.5.

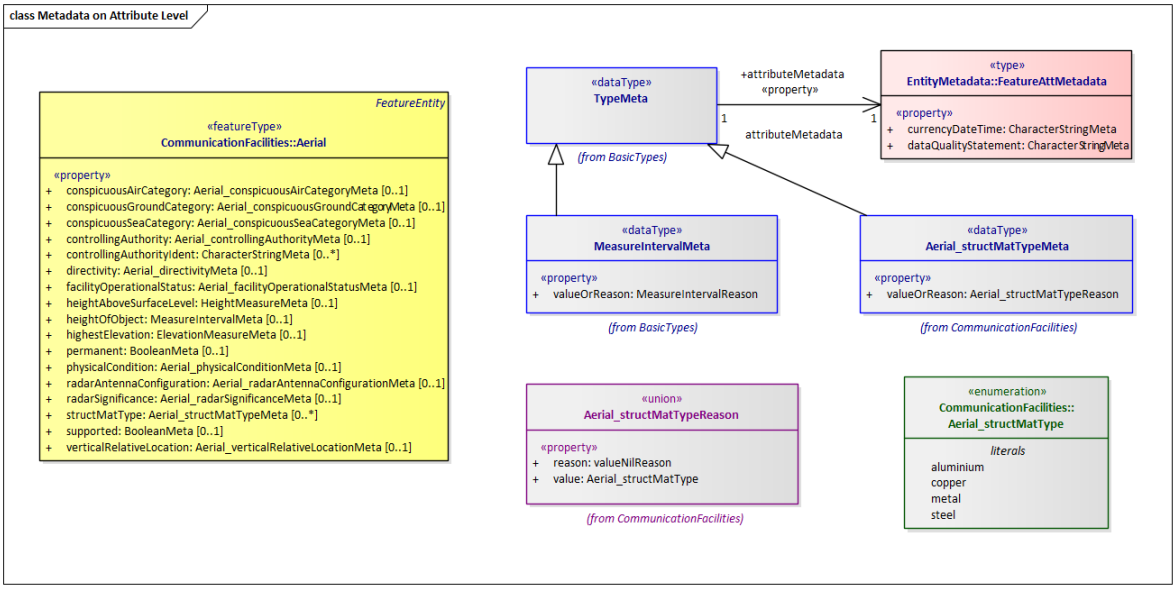

Figure 17: UML diagram depicting the use of actual values.

7.5 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.

7.5.1 Association of featureType to featureType

Associations between two or more featureTypes specify relationships that cannot be easily determined through spatial analysis. As noted in section 6.2.6, the use of associations to specify a spatial relationship between two or more featureTypes is generally avoided in DGIM. In certain cases, featureType associations are allowed, particularly when the relationship between features is not geospatial in nature.

Examples for allowed associations:

- A hospital-associated helipad located near, but not adjacent-to or contained-within a medical care facility.
- A LandTransportationWay that is part-of, but does not directly intersect a LandRoute.

Examples of disallowed associations:

- A PopulatedPlace contained-within an AdministrativeDivision
- A RunwayElement contained-within or intersecting-with a Runway

7.5.2 Associations of type to featureType

A type usually specifies non-spatial information. It can be associated to featureType to specify additional detailed information or specific domain related specification. A featureType can be associated with a unidirectional link to one or more types.

7.5.3 Associations of type to type

The use of associations to link two or more types is generally discouraged with DGIM. As of DGIF 3.0, complex chains of associated types are still known to cause technical challenges both in deriving profiles and subsets of the DGIM, and in implementing relatively large DGIF subsets in production environments using commercial off the shelf (COTS) software. Some progress has been made in overcoming these technical difficulties, so more complex type to type associations may be implemented in future releases of DGIF.

8 Content

The content of the DGIM is managed by the DGIWG Vector Models and Schema Team (VMST), which functions as both a configuration control body and development team.

Although the VMST is responsible for model development and innovation, this standard (DGIWG 205) establishes the basic rules and parameters for the DGIM.

As of DGIF 3.0, the DGIM primarily supports exchange of defence geospatial information. As a result, some elements of the DGIM model are not optimized for data collection or for representation via web services.

8.1 Basic Principles of Model Management

8.1.1 Generic Model

The DGIM should be remain sufficiently generic to support a general representation of real-world phenomena.

For example:

- DGIM contains a LandTransportationWay concept that describes a “way” or a distinct strip of land used for ground transportation. This eliminates the need to specify similarity or distinction between closely-related concepts (e.g. path, track, drove, trail, road, etc.) at a feature level. Additional attribution enables further specification and description of the physical characteristics thus avoiding the need to determine a key semantic distinction between functionally similar types of “ways.”
- DGIM also contains an InlandWater concept instead of Pond, Reservoir, WaterRace, etc. for the same reason. DGIM enables users to specify the existence and characteristics of inland water without necessarily determining the classification of water body. The DGRWI serves to point users from these various colloquial terms to the formal DGIM model term.

8.1.2 Physical Representation vs. Thematic Meaning

DGIM separates physical representation from the usage or meaning of a given phenomenon.

This principle is expressed in the DGIM in two sub-principles:

8.1.2.1 Temporo-spatial phenomena specified by multiple DGIM features

This sub-principle enables different aspects of a phenomena to be specified independently. For example, a city may be captured as a DGIM BuiltUpArea feature that specifies the static physical characteristic of the city (e.g. buildings, heights, materials, etc.). The same city may

also be captured as a DGIM PopulatedPlace feature that specifies the function and meaning (capital, trading centre and settlement type, etc.).

8.1.2.2 Distinction of Physical and Thematic Phenomena

This sub-principle enables the distinction between “physical” featureTypes such as Building, Dam or Forest and the “thematic” featureTypes (e.g. land route, conservation area, maritime caution area, etc) that may be associated with physical featureTypes or may lack a real-world physical expression

9 Metamodel

9.1 Overview

The metamodel of DGIM mainly consists of packages[3], entities/featureTypes, dataTypes of any kind and Enumerations.

9.2 Elements

9.2.1 Application Schema

Meta class: Package

The top package comprising the whole DGIM is of stereotype applicationSchema.

9.2.1.1 Properties

The information in Table 8 is attached to the applicationSchema.

Table 8: Information attached to applicationSchema

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
Standard UML properties and other fields						
name	"DGIM"	String	1	YES		String
Stereotype properties (tagged values)						
gmlPro-fileSchema	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/gmlSimpleFeaturePro-file/1.1.0/gmlsf.xsd (defined by GML 3.2)	URI	0..1	NO		URI

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

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
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 9 is attached to a bundle or leaf.

Table 9: Information attached to bundles and leaves

Item	Definition/Content	Data type	Multi-plicity	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

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
fullName	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 10 is attached to a featureType.

Table 10: Information attached to featureType

Item	Definition/Content	Data type	Multi-plicity	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)						
byValueProperty-Type	A Boolean value where TRUE creates an additional property type that enforces that the Entity Type is encoded in line with 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 Entity 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

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
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
geometry	The default allowed geometries 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,MS o"	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
securityClassification	The security classification that applies to this entity.	SecurityClassification	1	YES		SecurityClassification
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

9.2.4 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 to 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.

9.2.5 Type (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.

9.2.5.1 Properties

The information shown in Table 11 is attached to a type.

Table 11: 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)						
byValueProperty-Type	A Boolean value where TRUE creates an additional property type that enforces that the Type is encoded in line with 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

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
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
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

9.2.5.2 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 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.

9.2.6 Data Type (dataType)

Meta Class: UML dataType

A dataType represents the actual nature of a property.

9.2.6.1 Properties

The information shown in Table 12 is attached to a dataType.

Table 12: Information attached to dataType

Item	Definition/Content	Data type	Multi-plicity	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_Item-Status	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

9.2.6.2 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.

9.2.7 Union

Meta Class: UML Data Type

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

9.2.7.1 Properties

The information in Table 13 is attached to a union.

Table 13: Information attached to union

Item	Definition/Content	Data type	Multi-plicity	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_Item-Status	1	YES		RE_ItemSta-tus
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
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

9.2.7.2 Links

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

9.2.8 Property (Attribute)

Meta Class: UML Property

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

9.2.8.1 Properties

The following information shown in Table 14 is attached to a **property**.

Table 14: Information attached to property

Item	Definition/Content	Data type	Multi-plicity	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
inlineOrByRefer-ence	A value indicating how this Entity Attribute should be encoded in GML. One of: { 'inline', 'byReference', 'inlineOrByReference' }. (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. Applicable in specific derived representations, for example flattened representations. (ShapeChange extension)	Integer	0..1	YES		Integer

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
profiles	Comma separated list of identifiers that define in which profiles the Property is applicable. A parameter maxOccurs allows further specifications, e.g. "TM50(maxOccurs=2)" to overwrite maxOccurs for a specific profile. (ShapeChange extension)	String	0..1	YES		String
securityClassification	The security classification that applies to this Property.	SecurityClassification	1	YES		SecurityClassification
sequenceNumber	Unique index of this Entity Attribute within the Entity Type; may be used to tag derived products in order to ensure consistent Entity Attribute ordering. (defined by GML 3.2)	Integer	1	YES		Integer
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

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

9.2.9 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.

9.2.9.1 Properties

The information in Table 15 is attached to an enumeration.

Table 15: Information attached to enumeration

Item	Definition/Content	Data type	Multi-plicity	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

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
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_Item-Status	1	YES		RE_ItemSta-tus
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

9.2.9.2 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

9.2.10 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.

9.2.10.1 Properties

The information in Table 16 is attached to a codeList.

Table 16: 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_Item-Status	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

9.2.10.2 Links

A codeListCodelist is referenced by a dataType.

9.2.11 Enum

Meta Class: Property

An enum is a value within an enumeration.

9.2.11.1 Properties

The information shown in Table 17 is attached to an enum.

Table 17: Information attached to enum

Item	Definition/Content	Data type	Multi-plicity	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 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

9.2.11.2 Links

An enum belongs to one specific enumeration.

9.2.12 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.

9.2.12.1 Properties

The following information in Table 18 is attached to a property. (They are similar to the property described in 9.2.8)

Table 18: Information attached to property

Item	Definition/Content	Data type	Multi-plicity	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)						

Item	Definition/Content	Data type	Multi-plicity	Norm.	Generation	EA Datatype
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', 'inlineOrByReference' }. (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. Applicable in specific derived representations, 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 applicable. (ShapeChange extension)	String	0..1	YES		String
securityClassifi-cation	The security classification that applies to this Property.	Securi-tyClassifi-cation	1	YES		SecurityClas-sification
sequenceNumber	Unique index of this Entity Attribute within the Entity Type; may be used to tag derived products in order to ensure consistent Entity Attribute 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 encoding for this Schema and its contents. (ShapeChange extension)	String	1	YES		String

9.2.12.2 Links

A property belongs to a specific association.

9.2.13 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 (see Table 19).

Table 19: 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

10 Governance and Maintenance

The management of the DGIF, including the DGIM conforms to the governance process established by the DGIWG Vector Data Technical Panel (P1) and is executed by the VMST. See DGIWG 200 – DGIF Annex A for more information.

11 Quality Assurance

Proposed changes to the DGIM shall be verfied to ensure that the proposal conforms to the rules set out in this standard. It is the responsibility of the submitting party to perform this verification. The DGIM steward will also perform Quality Assurance (QA) checks during the DGIF management cycle. The steward will communicate any identified issues to the originator of the change proposal for resolution prior to the proposal being formally considered for integration into the standard.