



DGIWG 207

Defence Geospatial Real World Object Index (DGRWI)

Document type:	Standard
Document date:	28 November 2017
Edition:	2.0.0
Supersedes	This document supersedes DGIWG 204 Ed. 1.0.0, NATO Geospatial Real World Object Index (NGRWI), 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 Real World Object Index (DGRWI) as part of the Defence Geospatial Information Framework (DGIF).
Copyright:	<p>(C) Copyright DGIWG, some rights reserved - (CC) (By:) Attribution</p> <p>You are free:</p> <ul style="list-style-type: none">- to copy, distribute, display, and perform/execute the work- to make derivative works- to make commercial use of the work <p>Under the following conditions:</p> <ul style="list-style-type: none">- (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. <p>Any of these conditions can be waived if you get permission from the copyright holder DGIWG.</p> <p>Your fair use and other rights are in no way affected by the above.</p> <p>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/>)</p>

Contents

1 Scope	2
2 Conformance	2
3 Normative References	3
4 Terms, Definitions, Abbreviations & Acronyms	3
4.1 Terms & Definitions	3
4.2 Abbreviations & Acronyms	5
5 Logical Structure	6
5.1 Logical Metamodel	6
5.2 General Model Specifications	6
6 Content	8
6.1 Typical Content	8
6.2 Responsibilities	8
6.3 Basic Principles	8
6.4 Change Requests	9
6.5 Content Rules	9
6.6 Quality Assurance	9
Annex A - Applications of the Defence Geospatial Real World Object Index	10
A.1 Search and Find in the Model	10
A.2 Maintain Consistency	10
A.3 Assist with Modelling User Requirements	10
A.4 Mapping and Transformations	10
A.5 Synonyms and Language Issues	10
Annex B - Metamodel for the Defence Geospatial Real World Object Index	11
B.1 Introduction	11
B.2 DGRWI stereotypes	12
B.2.1 Real World Object	12
B.3 UML Diagram	14
Annex C - Real World Object Tuples and OCL Encoding	17
C.1 Tuples	17
C.2 OCL Encoding	18
C.3 OCL Syntax for Navigation over Associations	22
Annex D - Real World Object Change Types and Guidance	23
D.1 Addition	23
D.2 Supersession	23
D.3 Retirement	24
Annex E - Real World Object Content Rules	25
E.1 Real World Object Stereotype	25
E.1.1 AlphaCode (Class Name) [Mandatory]	25
E.1.2 Alias [Mandatory]	25
E.1.3 OCL Constraint: [Optional]	25

E.1.4 Notes [Optional].....	25
E.1.5 531 Code [Mandatory].....	26
E.1.6 English Term [Mandatory]	26
E.1.7 Profiles [Mandatory]	27
E.1.8 Significance [Mandatory]	27
E.1.9 Source [Optional].....	27
E.1.10 OCL Statement [Mandatory].....	27
E.1.11 OCL Comment [Mandatory].....	27
E.1.12 Description [Conditional].....	28
E.1.13 Status [Mandatory]	28
E.1.14 Date Accepted [Mandatory]	28
E.1.15 Date Retired [Conditional]	28
E.2 Representation Stereotype.....	28
E.2.1 Target Role (Mandatory)	28
Annex F - DGRWI Queries	29
F.1 Identify Next available RWO 531Code	29
F.2 QA Check Alphacode Entries	29
F.3 QA Check 531 Codes.....	30
F.4 QA Check English Term	30
F.5 QA Check Source.....	31
F.6 QA Check Description	31
F.7 QA Check Status.....	31
F.8 QA Check OCL	32
F.9 QA Check OCL Comment	32
F.10 QA Check OCL Statement	33
F.11 QA Check Significance.....	33
F.12 List of all Real World Objects linked to a given Entity.	34

List of Tables

Table 1: Normative References.....	3
Table 2: Definitions Applicable to this Standard	3

i. Submitting organizations

Nation	Organisation
United Kingdom	Joint Forces Intelligence Group (JFIG)
Estonia	Military Geographic Group, Estonian Defence Forces
Germany	Bundeswehr Geoinformation Centre

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
2015-01-01	0.1	DEU	All	Initial Template
2015-01-26	0.2	GBR	All	Initial Population of information
2015-01-28	0.3	GBR	All	Revision and rework
2015-12-09	0.4	GBR	All	Revision and rework. Consolidation of DGRWI documentation.
2016-01-29	0.9	GBR	All	Revision and rework. Consolidation of DGRWI documentation.
2017-06-01	0.91	GBR	All	Editorial Checks and Corrections
2017-10-05	1.0	GBR	All	Comment Resolution

iv. Future work

As the DGIF is expected to evolve and business and technical processes mature, this DGRWI specification will be reviewed and updated accordingly.

Introduction

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

The Defence Geospatial Feature Concept Dictionary (DGFCDD) provides the standardized semantics of all Concepts required by the Defence Geospatial Information Model (DGIM) and is the authoritative source for the model. It is accompanied by the Defence Geospatial Real World Object Index (DGRWI) which provides an Index of identified real world geospatial phenomena and a reference to their representation in the DGIM.

All these artefact of DGIF are maintained in the Defence Geospatial Information Framework Collaborative Modelling Environment (DCE).

The DGIM itself is a DGIF-wide logical model for geospatial vector data that is technology neutral. This Platform Independent Model (PIM) determines the syntactic structure and the DGFCDD provides standardized Feature Concepts, Attributes Concepts, Datatypes, Unit of Measures and Concepts for Enumeration Values.

As commonly used names or terms for real world phenomena may not be realised as explicit Feature Types in the DGIM, the DGRWI supports entry point usage of DGIF by providing an index and reference from Real World Phenomena terms to their representation in the DGIM.

For example a Real World Object (RWO) of **Quarry** is represented as a Feature Type *Extraction Mine* with an attribute of *Extraction mine type = Quarry*. Without the DGRWI, a user/implementer non-conversant with the DGIM may wrongly assume that commonly used names or terms for real world phenomena may not be realised as explicit Feature Types in the DGIM and that Quarry is not a represented feature in the defined schema, while in fact the concept is realised.

1 Scope

The scope of this standard is to provide information on the purpose and structure of data within the DGRWI as part of the DGIF suite of standards.

When real world phenomena or objects are modelled digitally, vector geometric entities consisting of points, curves and surfaces are used. To establish the relationships between these entities and the real world features/objects they represent, the concept of a Real World Objects is used,

Vector data is extracted, managed and provided within a defined schema and in a specified structure (in the form of a Feature Catalogue and/or a derived application schema). Within the DGIF, the DGIM contains the business rules and specifications used to describe the geospatial vector data model.

The DGFCF contains the concepts valid for the DGIM. The DGFCF presents an abstraction of reality as a defined classification of phenomena. The basic level of classification is the feature type.

Even though the DGIM contains standardised feature types and attribute concepts from the DGFCF, a Real world Object (RWO) may not be directly (explicitly) referenced in the Feature Catalogue/Model but may be represented by another feature-attribute combination elsewhere in the DGIM.

For example, while a RWO **Bridge** can be easily found as a *Bridge* feature type in the DGIM, the RWO **Quarry** is modelled/defined as Feature Type *Extraction Mine* with an attribute of *Extraction mine type = Quarry*.

The DGRWI has been developed to record these RWOs and their DGIM reference. For example the DGRWI provides a searchable reference *Quarry* and its associated DGIM representation. The DGRWI allows the identification and retrieval of RWOs without the need for a detailed knowledge of the DGIM and DGFCF.

In addition common synonyms, when identified, are included to further simplify the extraction process and make the index more user-friendly.

For example searching for RWOs *Pit, Mining Plant, Quarry, Colliery or Mine* would all locate the feature type *ExtractionMine*

0 provides examples and further information on the potential applications of the DGRWI.

2 Conformance

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

No explicit conformance to the DGRWI is defined.

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.

Table 1: Normative References

Standard or Specification
DGIWG - 205 - Defence Geospatial Information Model (DGIM)
DGIWG - 207 - Defence Geospatial Real World Object Index (DGRWI)
DGIWG - 208 - DGIF Exchange Formats for Vector Data - Part-1 GML
DGIWG - 114 - DGIWG Metadata Foundation

4 Terms, Definitions, Abbreviations & Acronyms

4.1 Terms & Definitions

The terms and definitions specific to this standard are given in Table 2.

Table 2: Definitions Applicable to this Standard

Term	Definition
Attribute	A characteristic of a feature.
Application Schema	Conceptual schema for data required by one or more applications [ISO 19101].
Conceptual Model	Model that defines concepts of a universe of discourse [ISO 19101].
Datatype	Specifies how the value of an Attribute shall be abstractly represented and consists of one or more fields, each capturing an aspect of information required to completely specify a value in the domain of the datatype. A simple datatype consists of a single field containing a primitive data value (e.g. a real number); a complex datatype consists of multiple fields, at least one of which contains a data value and others may contain metadata about the data value(s).
Enum	Values that are members of the domain of a specific enumerated datatype. These listed values are often referred to simply as "enums".
Feature	Abstraction of real world phenomena [ISO 19101]. NOTE A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant.

Term	Definition
Feature Attribute	<p>Characteristic of a feature [ISO 19101].</p> <p>NOTE 1 - A feature attribute may occur as a type or an instance. Feature attribute type or feature attribute instance is used when only one is meant.</p> <p>NOTE 2 - A feature attribute type has a name, a data type and a domain associated to it. A feature attribute instance has an attribute value taken from the domain of the feature attribute type.</p>
Feature Catalogue	Catalogue containing definitions ... of the feature types ... occurring in one or more sets of geographic data [ISO 19110].
Geographic Data	<p>Data with implicit or explicit reference to a location relative to the Earth.</p> <p>NOTE Geographic information is also used as a term for information concerning phenomena implicitly or explicitly associated with a location relative to the Earth.</p>
Physical Quantities	A set of physical quantities that characterize the properties of a phenomenon, body, or substance, where the property has a magnitude that can be expressed as a number (physical value) and a reference quantity - referred to as a "unit of measure".
Real World Object	An existing geographic (or geospatial) occurrence whose characteristics can be described/identified e.g. A Wooden Bridge, A Mosque, A Divided Highway.
Real World Object Tuple	A three element Feature Type-Attribute-Value combination used to describe a Real World Object (RWO).
Units of Measure	A set of units of measure, organized by physical quantity, where a unit of measure is a predefined amount of the concerned physical quantity (for example: a metre "of length" or kilogram "of mass").
Universe of Discourse	View of the real or hypothetical world that includes everything of interest [ISO 19101]

4.2 Abbreviations & Acronyms

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

DCE	DGIF Collaborative Modelling Environment
DFDD	DGIWG Feature Data Dictionary
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
GIRD	Geospatial Information Requirement Description
ISO	International Organization for Standardization
NATO	North Atlantic Treaty Organization
JGSWG	(NATO) Joint Geospatial Standards Working Group
NGRWI	NATO Geospatial Real World Object Index
OCL	Object Constraint Language
PIM	Platform Independent Model
QA	Quality Assurance
RWO	Real World Object
TS	Technical Specification
UML	Unified Modelling Language
VMST	DGIWG Vector Model and Schema Team

5 Logical Structure

The DGRWI provides indexing from names or terms of real world phenomena to their representation in the DGIM.

The index is implemented in UML in the DCE. The underlying metamodel of the DGRWI is defined within the DCE.

5.1 Logical Metamodel

In general the metamodel is organized as follows:

There is one basic UML element: **Real World Object (RWO)**.

The RWO element is connected to one or more relevant *Feature Types* or *Types* within the DGIM using the *Representation* link. The RWO provides a definition based on the content of the connected elements.

Each RWO contains the basic information needed for maintenance and management following the principles of item registration by ISO 19135 – Geographic information - Procedures for item registration. More details on the management can be found in HBK-13-047-Change Management for DGIF (CM-DGIF).

The detailed description of DGRWI metamodel (e.g. all used stereotypes and tagged values) is provided at Annex B.

5.2 General Model Specifications

Each entry in the DGRWI requires specific information as follows (additional detailed information on content is at Annex E):

1. **(Class) Name:** The UML class name represents the alphaCode; a unique alphanumeric value that may be used to designate the RWO, e.g. *CanalTunnel*. The alphaCode is also called the Primary Code and represents a human readable name that can be longer than 25 characters. It should be used as primary identification although as some information systems may be not able to use identifiers of this length, the Secondary Code may be used.
2. **Alias:** The alias informally represents the Secondary Code; 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, e.g. *RWO_00578*. The Secondary Code should only be used when the alphaCode cannot be used in the system environment.
3. **Status:** The status of the element following ISO 19135 specifications. Allowed values are *valid*, *notValid*, *retired*, *superseded*. Only valid items are part of the current model.
4. **Date Accepted:** The date when the element was accepted to be a valid element with the DGRWI.
5. **Date Retired:** The date when the element was retired from the DGRWI and became *retired* or *superseded*.

- 6. Object Constraint Language (OCL) Constraint:** The OCL constraint describing the (attributive) condition under which the feature type is representing the RWO. To support Unified Modelling Language (UML) implementation, each RWO has an OCL encoding that implements it, e.g.

```
inv:entity1.transportationSystemType.valueOrReason.value=TunneltransportationSystemType::canal
```

- 7. Notes:** These are general notes about the entry. These are used to contain human readable information about the DGIM Elements that the RWO describes. The format is:

-- Representation: --

Tunnel with transportationSystemType = canal

This is a reformatted version on the content of the OCL Comment.

- 8. English Term:** A unique phrase or term for the RWO. This should be:

- A widely understood English term for the concept, e.g. **Coal Mine**. This may be a formal recognized term from a relevant source or a common colloquial used English term.
- A designated and/or commonly used Operational, Military, Legal or User Community focused term relevant to DGIF content.
- An alternative/synonymous English term to an existing concept, e.g. **Colliery**.
- A descriptive term derived from the Feature/Attribute information in the DGIM, e.g. **Coal Mining Facility**. This may result in a term that is not 'regular' English.

- 9. 531 Code:** A unique alphanumeric value that formally represents the secondary code. (See Alias)

- 10. Profiles:** Information allowing identifying the profile usage of the concept in DGIF applications/specifications. The impact of any change proposed on this concept can be identified and communicated.

- 11. Significance:** An indication of the significance of the object to the user. Allowed values are *Modelling*, *Basic*, *Extended*. These Values are explained at Annex E.1.8

- 12. Source:** The source for the element to allow additional research in the management process. This may include a reference to a formal glossary of terminology.

- 13. OCL Statement:** The pure OCL statement for the constraint.

Example:

```
inv:entity1.transportationSystemType.valueOrReason.value=TunneltransportationSystemType::canal
```

(The OCL statement and the OCL comment are normative and are combined within the constraint of the RWO class.)

- 14. OCL Comment:** A human readable definition of the OCL statement

Example:

```
/* Tunnel has attribute transportationSystemType with value canal */
```

(The OCL statement and the OCL comment are normative and are combined within the constraint of the RWO class.)

15. Description: A general comment or additional information about the RWO.

6 Content

6.1 Typical Content

The RWO Index is a registry of terms of geographical phenomena in the English language and relevant to the DGIF Model content and user community. The terms in the DGRWI can be formally recognized/published terms from a relevant source or common colloquial English expressions.

Generally the type of terms included will be common nouns although this is not a finite rule. **'Forest'** and **'Facility'** will be included but **'Dense Forest'** and **'Non Usable Facility'** generally would not (unless identified as a distinct and required Noun Phrase, e.g. **'All Weather Hard Surface Road'**). Other terms in the form of a descriptive noun, e.g. **Oak Forest, Mangrove Swamp, Fresh Water**, should be assessed on their merits.

In all cases, in determining appropriate DGRWI terms and synonyms the potential use by an end user should always be in mind. The 'Significance' value enables types of RWO to be distinguished.

6.2 Responsibilities

The DGRWI is maintained using the DCE. The DCE allows change and maintenance to be performed by each member of the DGIWG Vector Modelling and Schema Team (VMST) through an established change proposal process.

The content of the DGRWI are managed by the Maintenance Team based on technical and content assessment of these Proposals, followed by an official voting by each DGIWG member nation. The DGRWI Custodian governs the content development.

Content changes and development can result through internal Maintenance Team/Custodian initiative or as a result of external user and client (for example NATO) requirement tasked directly through National maintenance team representatives or indirectly through DGIWG Requirements Group. National or Client VMST members are expected to directly add change requests for DGRWI to DCE. DGIF Authorised National Representatives (ANRs) will be asked to vote on change requests in line with VMST procedures.

Content changes for the DGRWI can be independent to or in conjunction with DGIM and DGFCF Changes.

6.3 Basic Principles

The main information maintained in the DGRWI is the English Term which is connected via a Representation to one or more Entity-Attribute-Value tuples in the DGIM.

It is necessary to identify the appropriate tuples and their OCL encoding prior to making a change request.

The concept of tuples and their OCL encoding is explained in Annex C.

6.4 Change Requests

All Change Proposals to the DGRWI are made within the DCE.

The process for DCE Change Proposals is outlined in HBK-WD-13-044-Change Management for DGIF (CM-DGIF)

There are three basic change operations for RWOs; Addition, Supersession, and Retirement. These are described, with guidance, at Annex D.

6.5 Content Rules

Proposals for RWOs added to DCE should conform to the content rules as set out at Annex E.

6.6 Quality Assurance

The Proposer of changes to the DGRWI has responsibility to ensure that the content of the proposal conforms to the rules set out in this document. See Annex E.

The DGRWI Custodian will perform additional Quality Assurance (QA) checks

Details of the DGRWI QA Checks are found at Annex F.

Annex A- Applications of the Defence Geospatial Real World Object Index

The DGRWI may be used for the following applications:

A.1 Search and Find in the Model

The structure and optimisation of the DGIM may imply that some Real World Phenomena, from a user's perspective, are 'hidden' or not obviously immediately apparent. For example a user looking for a 'Road' may be perplexed as to why a feature called Road is not in the model. (Roads are represented in the Model but are covered by the *LandTransportationWay* feature concept). However **Road** (and variations of that term) will be searchable in the DGRWI.

The DGRWI also allows the identification and retrieval of RWOs without having to have a detailed knowledge of the DGIM and DGFCF.

Users can identify Real World Phenomena that are not directly (explicitly) reflected by the model (e.g., **Gold Mine** or **Desert Track**).

A.2 Maintain Consistency

As inconsistencies in the DGIM need to be avoided, modellers may want to check if real phenomena are not represented by more than one combination of model elements. For example, the model must not define more than one representation of a **Church**. If a church is defined by the attribute *Feature Function code* on the feature type *Building* then it should not also be represented by another modelling element, like the feature type *Church*.

A.3 Assist with Modelling User Requirements

Requirements are often not expressed in technical terms. For instance, a user may need to model a lake. By searching for **Lake** in the DGRWI, one is able to connect the user requirement to the feature type *Inland Waterbody* associated to the attribute *Inland Water Type*.

A.4 Mapping and Transformations

If one is mapping features between two different models, both having their own RWO index, searching by their real world terms may help find the corresponding modelling constructs, without having to deeply analyse the data models.

A.5 Synonyms and Language Issues

The terminological definition of RWOs allows the use of synonyms and technical terminology, which in turn caters for the variation in the use of the English language, colloquialisms and ambiguities. This also makes translations easier.

For example searching for RWOs; **Pit, Mining Plant, Quarry, Colliery or Mine** would all locate the feature type *ExtractionMine*.

Annex B - Metamodel for the Defence Geospatial Real World Object Index

B.1 Introduction

The following chapter is not necessary for the usage of the DGRWI. It is not required to understand these concepts in order to use it. The chapter briefly describes the metamodel as it is used in the DCE. For more detailed information document HBK-'WD-13-044-Defence Geospatial Information Framework Collaborative Modelling Environment' is recommended.

The DGIF metamodel (See Figure B.1) consists of many stereotypes and their associated properties.

These stereotypes were created to cover specific requirements for ISO compliance. For example, to either follow the stereotypes as defined within ISO TC 211 or even more importantly to represent all specific items in the artifacts of DGIF, DGIM, DGFCF and the DGRWI. Additional stereotypes have been introduced to cover the Geospatial Information Requirement Description (GIRD).

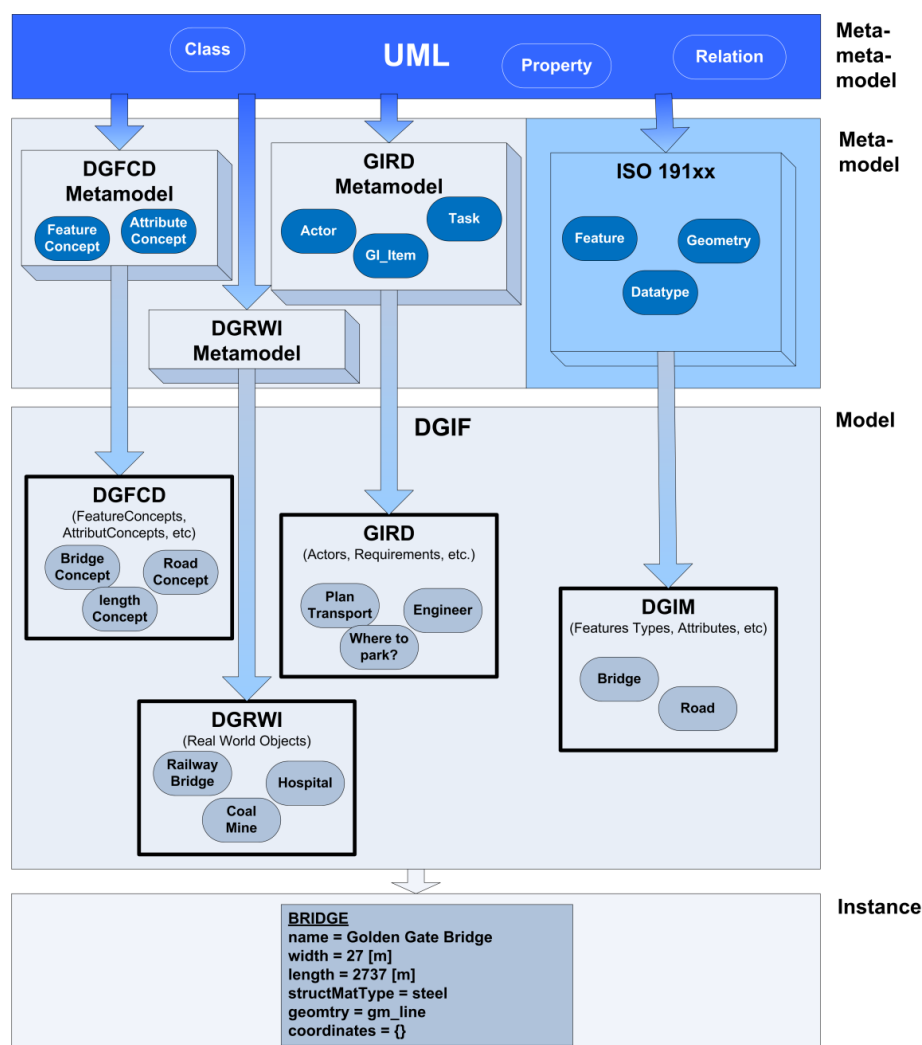


Figure B.1 DGIF metamodel

B.2 DGRWI stereotypes

The DGRWI consists of the following stereotype:

The stereotype description consists of information about the Meta Class it extends, the properties it owns, the links to other stereotypes that may exist, and an example.

The properties are described using following terms:

- **Item:** The name of the stereotype's property (either a standard UML property or a specially created property for this stereotype).
- **Definition/Content:** The description of the property.
- **Datatype:** The name of a general datatype (e.g. Text or Real).
- **Multiplicity:** Defines the allowed number how often a property can be used.
- **Normative:** A Boolean field defining if it is mandatory for a property to be populated.
- **Generation:** A note about the source of property content if it is not mandatory. Usually properties that have information about generation are populated automatically from other properties.
- **DCE Data type:** The datatype that is used in the DCE. In case of big text fields (>255 characters) a specific datatype called "Memo" is used.

B.2.1 Real World Object

Meta Class: Class

This stereotype represents a **realWorldObject** as defined in the DGRWI.

B.2.1.1 Properties

The following information is attached to a **realWorldObject**:

Item	Definition/Content	Data type	Multi- plicity	Norm.	Gen	DCE Datatype
Standard UML properties and other fields						
(class) name	The alphaCode as defined in the DGRWI.	String	1	YES		
alias	The 531-Code as defined in the DGRWI	String	1	YES		
constraints	The OCL constraint describing the (attributive) condition under which the feature type is representing this RWO. It is combined from the OCL Statement and the OCL comment	String	1	NO	from OCL Statement and OCL comment	
notes	A human readable statement about the DGIM Elements that the RWO describes.	Memo	1	NO	Version of OCL comment	
Stereotype properties (tagged values)						
531Code	The 531-Code as defined in the DGRWI consisting of the String "RWO_" followed by a five digit number, for example "RWO_04325".	String	1	YES		String
englishTerm	The English term designating the phenomenon	String	1	YES		String
profiles	Comma separated list of identifiers that define in which profiles the RWO is applicable.	String	0..1	YES	see chapter 11	String
significance	A value describing the significance of the object regarding the usage in a deliverable (default, basic, extended)	significanceList	1	YES		significanceList
source	The source from which the name is originating	String	0..* (separated by semicolon)	NO		String
oCLStatement	The pure OCL statement for the constraint.	String	1	YES		Memo
oCLComment	A human readable definition of the OCL statement as an OCL Comment	String	1	YES		Memo
description	A general comment about the RWO.	String	0..1	NO		Memo
status	The status of the RWO as defined in ISO 19135 ('valid', 'notValid', 'retired', 'superseded')	RE_ItemStatus	1	YES		RE_ItemStatus

Item	Definition/Content	Data type	Multi- plicity	Norm.	Gen	DCE Datatype
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

B.2.1.2 Links

A **realWorldObject** is connected to a *featureType(s)* and/or *type(s)* to represent the OCL Statement in a graphical way. These links are consistent with the Real World Object Tuple(s).

The normative information about the connection between a **realWorldObject** and an Entity-Attribute-Value Tuple is the oCLstatement and the oCLcomment. The OCL constraint is derived from these fields. The connector **representation** is normative.

B.2.2 Representation

Meta class: Dependency

A **representation** connects a **realWorldObject** with its represented **Feature Type** or **Type**.

Source Class	Source Role	Source Mult	Target Class	Target Role	Target Mult	Dir
realWorldObject		0..*	Entity	entity[no]	1..3	->

At this point in time a **representation** does not have specific tagged values attached to it. The target role is designated as “entity” with a consecutive number depending on the connections, for example “entity1” for an RWO that is only connected to one type.

B.3 UML Diagram

Figure 2.3 Shows a UML Representation of the DGRWI Metamodel

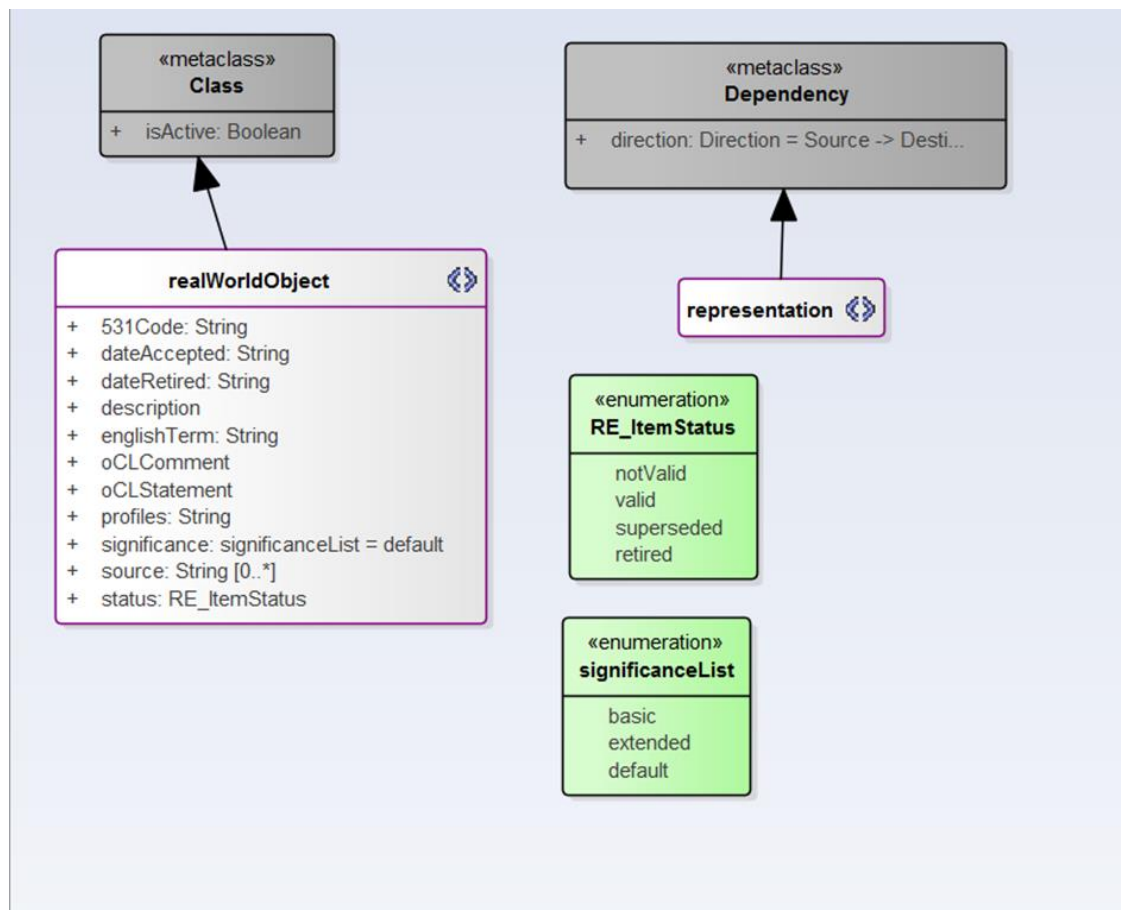


Figure 2.3 UML Representation of the DGRWI Metamodel

B.3.1 Example

Figure B.2.4 shows an example of Stereotype used in the Model:

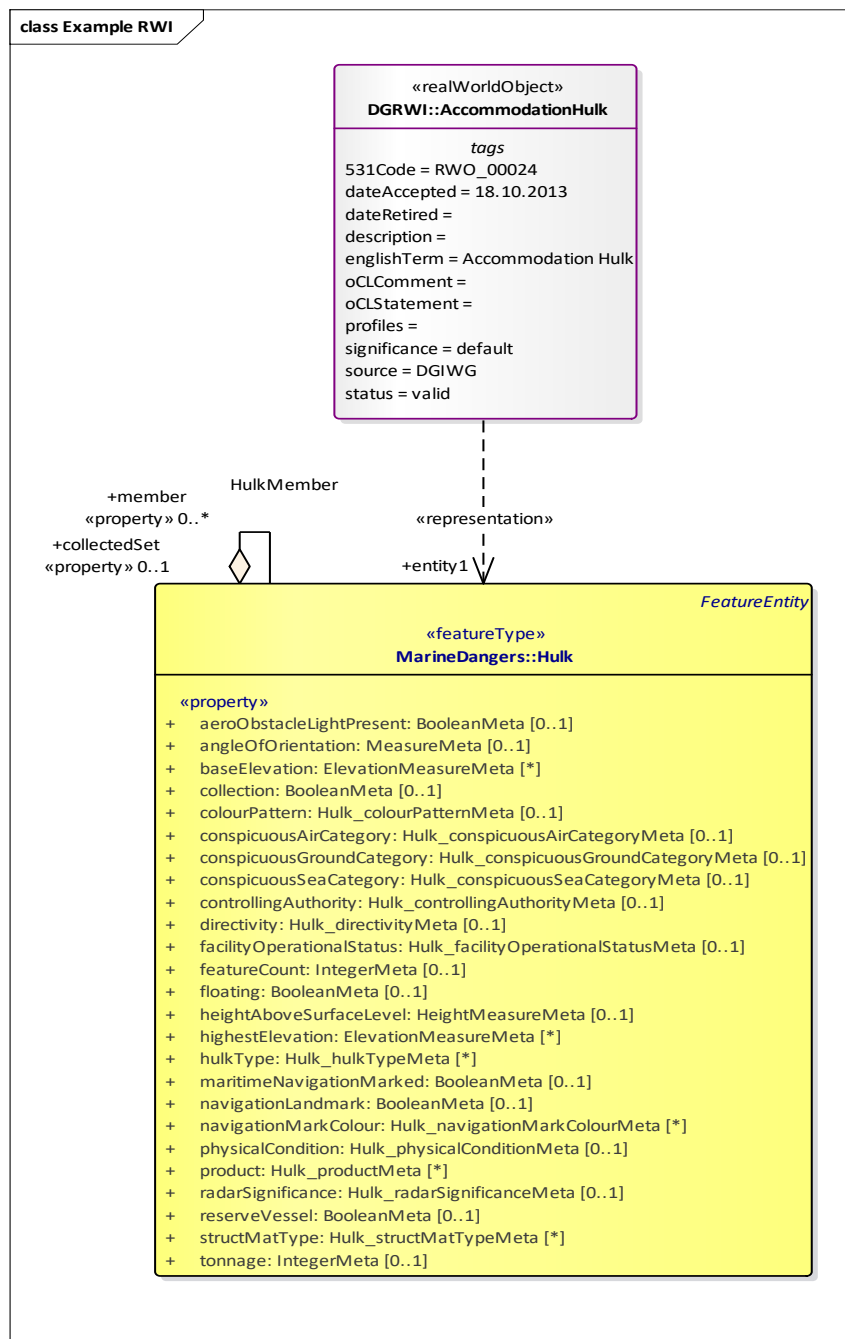


Figure B.2.4

Annex C- Real World Object Tuples and OCL Encoding

C.1 Tuples

A Tuple is Feature/Type-Attribute-Value combination from the DGIM that is required to represent a RWO.

A RWO must be represented by one tuple. It is possible to combine multiple tuples into one tuple to define more complex distinct objects. In the case of using more than one tuple it is understood to be a combination ("and") and not as alternatives ("or"). In theory any number of Tuples can be designated but in practice no more than three is recommended. Additional tuples can be for different attributes from the same Feature Type or Type or from an independent Feature Types or Types.

Examples:

RWO "Canal"

- DGIM Tuple 1: Feature Type = *Canal* (no Attribute or Value)

RWO "Irrigation Ditch"

- DGIM Tuple 1: Feature Type = *Ditch*; Attribute Type = *ditchFunction*; Value = *irrigation*

RWO "Historic Palace"

- Tuple 1: Feature Type = *Building*; Attribute Type = *featureFunction*; Value = *palace* AND
- Tuple 2: Feature Type = *Building*; Attribute Type = *historicSignificance*; Value = *historic*

RWO "Buddhist Monastery Facility" (and its synonym RWO "Lamasery")

- Tuple 1: Feature Type = *Facility*; Attribute Type = *featureFunction*; Value = *religiousActivities* AND
- Tuple 2: Type = *ReligiousInformation*; Attribute Type = *religiousDesignation*; Value = *Buddhism* AND
- Tuple 3: Type = *ReligiousInformation*; Attribute Type = *religiousFacilityType*; Value = *monastery*

RWO "Fresh Water Well"

- Tuple 1: Feature Type = *WaterWell* AND
- Tuple 2: Type = *WaterResourceInformation*; Attribute Type = *waterType*; Value = *fresh*

C.2 OCL Encoding

All RWOs require an OCL encoding. The identified tuple values and types are used to define the appropriate OCL Statement.

The Tuples are presented as OCL Statements within the DCE. An OCL Statement is a pure OCL encoding of the relevant tuple(s). An OCL Comment is a standardised human readable version of the OCL Statement. Care should be taken to ensure the logic of this statement when read as English is consistent with the concept (care with 'and', 'or' statements; introduce 'where' and 'if' etc. as required)

The OCL Statement must be written to the RWO Tagged Values OCLStatement field as well as the OCL Constraint field.

It is important to identify the Attribute type e.g. Enumeration, IntegerMeta, etc, used in a tuple as this affects the OCL Encoding.

The OCL constraint of a RWO in DGRWI is always referring to one or more feature type and/or type in the DGIM. The aim of the OCL is to identify how a RWO is represented by feature-attribute or feature-association combination in the DGIM. For example: RWO MetalBridge refers to feature type Bridge in DGIM with property verticalConstMaterial = metal. To indicate which DGIM featureType the OCL is referring to, the target name of connecting representation link is used.

As all representations between RWOs and corresponding feature Types in DGIM have the target name entity1, this name must be used. If there are two feature Types/types involved the target name would be entity2, entity3, etc.

The following provides syntax and examples of OCL Encoding for Tuples with various common attribute types. (This is not a definitive list for all types. If the Attribute Type is not listed below advice should be sought.)

C.2.1 Tuples that contain no attribution (Type only)

OCL Syntax:

inv: self.entity1.ocllsKindOf(Type**)**

Examples:

RWO "Canal"

- DGIM Tuple 1: Feature Type = *Canal* (no Attribute or Value)
- OCL Statement = **inv: self.entity1.ocllsKindOf(Canal)**
- OCL Comment = */* Canal */*

C.2.2 Tuples using attributes with enumerations

OCL Syntax:

inv: entity1.propertyName**.valueOrReason.value = **enumerationName::enum****

Examples:

RWO "Irrigation Ditch"

- DGIM Tuple 1: Feature Type = *Ditch*; Attribute Type = *ditchFunction*; Value = *irrigation*
- OCL Statement
inv:
entity1.ditchFunction.valueOrReason.value=Ditch_ditchFunction::irrigation
- OCL Comment
/ Ditch has attribute ditchFunction with value irrigation */*

RWO “Road”

- DGIM Tuple 1: Feature Type = *LandTransportationWay*; Attribute Type = *roadwayType*; Value = *road*
- OCL Statement :
inv: entity1.roadwayType.valueOrReason.value = LandTransportationWay_roadwayType::road
- OCL Comment
/ TransportationWay has attribute roadwayType with value road */*

RWO “Historic Palace”

- Tuple 1: Feature Type = *Building*; Attribute Type = *featureFunction*; Value = *palace* AND
- Tuple 2: Feature Type = *Building*; Attribute Type = *historicSignificance*; Value = *historic*
- OCL Statement
inv:
entity1.featureFunction.valueOrReason.value=Building_featureFunction::palace and
entity1.historicSignificance.valueOrReason.value=Building_historicSignificance::historic
- OCL Comment
*/*Building has attribute featureFunction with value palace and Building has attribute historicSignificance with value historic */*

RWO “Buddhist Monastery Facility” (and its synonym RWO “Lamasery”)

- Tuple 1: Feature Type = *Facility*; Attribute Type = *featureFunction*; Value = *religiousActivities* AND
- Tuple 2: Type = *ReligiousInformation*; Attribute Type = *religiousDesignation*; Value = *Buddhism* AND

- Tuple 3: Type = **ReligiousInformation**; Attribute Type = **religiousFacilityType**; Value = **monastery**
- OCL Statement

inv:
entity1.featureFunction.valueOrReason.value=Facility_featureFunction::religiousActivities and
entity2.religiousDesignation.valueOrReason.value=ReligiousInfo_religiousDesignation::buddhism and
entity2.religiousFacilityType.valueOrReason.value=ReligiousInfo_religiousFacilityType::monastery
- OCL Comment

/ Facility has attribute featureFunction with value religiousActivities and ReligiousInfo has attribute religiousDesignation with value buddhism and ReligiousInfo has attribute religiousFacilityType with value monastery */*

C.2.3 Tuples using attributes with type MeasureMeta

OCL Syntax:

*inv: entity1. **propertyName**.valueOrReason.value = **numericValue***

Example:

RWO “**Small Area Forest**”

- Tuple 1: Feature Type = **Forest**; Attribute Type = **area**; Value = **<10000**
- OCL Statement

inv: entity1.area.valueOrReason.value < 10000
- OCL Comment

/ Forest has attribute area with value <10000 */*

C.2.4 Tuples using attributes with type IntegerMeta

OCL Syntax:

*inv: entity1. **propertyName**.valueOrReason.value = **numericValue***

Example:

RWO “**Two Storey Building**”

- Tuple 1: Feature Type = **Building**; Attribute Type = **floorCount**; Value = **2**
- OCL Statement

inv: entity1.floorCount.valueOrReason.value=2
- OCL Comment

/ Building has attribute floorCount with value 2 */*

C.2.5 Tuples using attributes with type BooleanMeta

OCL Syntax:

inv: entity1. **propertyName.valueOrReason.value=**true/false****

Example:

RWO “**Temporary Checkpoint**”

- Tuple 1: Feature Type = *Checkpoint*; Attribute Type = *permanent*; Value = *False*
- OCL Statement
inv: entity1.permanent.valueOrReason.value=false
- OCL Comment
/ Checkpoint has attribute permanent with value false */*

3.2.1 Tuples using attributes with type CharacterStringMeta

OCL Syntax:

inv: entity1. **propertyName.valueOrReason.value=**'Character String'****

Example:

RWO “**US State**”

- DGIM Tuple 1: Feature Type = *AdministrativeDivision*; Attribute Type = *administrativeUnitTypeName*; Value = *US State*
- OCL Statement
inv: entity1.administrativeUnitTypeName.valueOrReason.value='US State'
- OCL Comment
/ AdministrativeDivision has attribute administrativeUnitTypeName with value US State */*

C.2.2 Tuples using combination of types

OCL Syntax:

Appropriate syntax for each tuples attribute type joined by ‘and’

Examples:

RWO “**Fresh Water Well**”

- Tuple 1: Feature Type = *WaterWell* AND
- Tuple 2: Type = *WaterResourceInformation*; Attribute Type = *waterType*; Value = *fresh*
- OCL Statement
inv: self.entity1.ocllsKindOf(Canal) and entity2.waterType.valueOrReason.value=WaterResourceInfowaterType::fresh
- OCL Comment

*/*WaterWell where WaterResourceInfo has attribute waterType with value fresh */*

C.3 OCL Syntax for Navigation over Associations

If a RWO is defined by the existence of the association between feature types/types, the syntax would be:

inv: entity1.associationTargetName2->notEmpty()

where entity1 is association target name from the RWO to Feature Type 1 and associationTargetName2 is association target name from Feature Type 1 to Feature Type 2

Example:

RWO “**Capital**” (refers to the Populated Place that is a capital of a Geopolitical Entity)

- Tuple 1 : Feature Type 1 = *PopulatedPlace*
- Tuple 2: Feature Type 2 = *GeopoliticalEntity*
- Association from Feature Type 1 to Feature Type 2, Association Target Name = isCapitalOf
- OCL Statement
inv: entity1.isCapitalOf->notEmpty()
- OCL Comment
*/*PopulatedPlace has association to GeopoliticalEntity with target name isCapitalOf */*

Annex D - Real World Object Change Types and Guidance

D.1 Addition

The addition of a RWO semantically means to propose a new real world term that does not currently exist in the DGRWI and can be identified in the model.

This could be a new term for a combination of model elements that does not have an existing defined RWO or a new alternative/synonym term for an existing RWO. The term could be a formally used term from a recognised source or which is colloquially used within a particular environment.

The following guidance should be taken into account:

1. The proposer should identify the tuples required to describe the Proposed RWO in the model.

Although there is no limit to the number of tuples allowed to describe a RWO, it is recommended that overly complex combinations are avoided as they become harder to encode. If more than three tuples are required to identify the Object then the Proposer should consider if; a) the Object can be simplified, b) the Object can be broken into more generic Objects, or c) determine if the Object is required at all.

2. If the object cannot be identified in the model then advice should be sought from the VMST team. This may require DGCFD or DGIM change proposals.
3. If the proposer is planning to propose changes to the DGFCF or DGIM it is recommended that these proposals are done before or concurrent to DGRWI Proposals. If possible new RWOs should be determined and proposed within the same voting cycle as DGIM Changes, although new RWOs can be proposed independently at any time.
4. The Proposer should populate the fields for the proposed DGRWI Object. All fields that are classified as mandatory shall be populated, all fields classified as optional should be populated if possible. The Proposer should conform to all content rules as defined in Annex E
5. The proposer should consider or research synonyms for the name of the proposed RWO. If valid, commonly used, or valuable synonyms exist then additional unique RWO entries should be added to the Proposal.

D.2 Supersession

The supersession of a RWO is performed if; a) the code and therefore the English term are changed, or b) the represented model entity has changed. This results in a new valid RWO being proposed and the existing one superseded.

The following guidance should be taken into account:

1. If the OCL constraint is changed due the change from one Tuple to another the RWO should be superseded. OCL Statement and OCL Comment have to be copied in the OCL Constraint. The Representation has to be set to a new Entity if needed.

2. If the English Term is changed in the way that the alphaCode has to be changed, supersession has to be performed. The 531-Code has to change if the alphaCode is changed.

D.3 Retirement

The retirement of a RWO means a reduction of the model. A retired and therefore not valid RWO represents a thing or a phenomenon that is not covered by the model anymore.

The following guidance should be taken into account:

1. Before retiring a RWO it has to be verified that the RWO is not covered by a more generic Element within the model. For example if the RWO "Highway" is not covered by the model explicitly by an attributive information anymore, the phenomenon "Highway" is still covered by the more general model element "Road" and shall therefore not be retired by superseded.
2. Before retiring a RWO it shall be verified that the RWO is not used in models that has to be maintained for legacy reasons.

Annex E - Real World Object Content Rules

The following provides details for the rules and guidance for the population of DGRWI items:

E.1 Real World Object Stereotype

E.1.1 AlphaCode (Class Name) [Mandatory]

Format:

The AlphaCode will be denoted in CamelCase denotation starting with a capital letter and reflecting the English Term of the concept, for example, *Bridge*, *InlandWater*, or *FloodControlStructure*. As such it should be human readable.

Uniqueness:

The (class) name is a unique AlphaCode for each RWO. The AlphaCode should usually be a camel case version of the English Term.

Special characters:

Certain characters should not be used:

- There shall be no AlphaCode beginning with a non-alphabetic character.
- There shall be no diacritics in AlphaCodes.
- A range shall be expressed with the term "to".
- A hyphen ("-") and parenthesis ("(", ")") shall be removed.
- For objects whose names are naturally numeric consider adding a prefix.

E.1.2 Alias [Mandatory]

The Alias is consistent with the 531Code (see below).

E.1.3 OCL Constraint: [Optional]

The OCL constraint describes the (attributive) condition under which the Feature Type is representing the RWO. To support UML implementation, each RWO has an Object Constraint Language (OCL) encoding that implements it.

Generation:

This is a concatenation of the OCL Statement (E.1.10) and the OCL Comment (E.1.11). The OCL statement and the OCL Comment are normative and have to be copied in the OCL Constraint.

E.1.4 Notes [Optional]

General Notes or comments about the entry. These are used primarily to contain human readable information about the DGIM Elements that the RWO describes.

The format is:

-- Representation: --

Tunnel with transportationSystemType = canal

This is a reformatted version on the content of the OCL Comment.

E.1.5 531 Code [Mandatory]

The 531 Code consist of the prefix RWO followed by a five-digit unique number, e.g. RWO_12345. It is not allowed to have two RWOs with the same 531Code.

A Query has been set up in DCE to determine the next available Code available. See 'EDT-15-023-AU-Queries Documentation' for details. This query can also be found at Annex F.

E.1.6 English Term [Mandatory]

Source:

Consideration should be given to the terminology a user may use to search for a concept.

- A widely understood English term for the concept, e.g. **Coal Mine**. This may be a formal recognized term from a relevant source or a common colloquial used English term.
- A designated and/or commonly used Operational, Military, Legal or User Community focused term relevant to DGIF content.
- An alternative/synonymous English term to an existing concept e.g. **Colliery**.
- A descriptive term derived from the Feature/Attribute information in the DGIM e.g. **Coal Mining Facility**. This may result in a term that is not 'regular' English. (The term should be made to read as 'regular' as possible e.g. **Double Track Railway** is a more 'natural' phrase than **Railway Double Track**)

In all cases, consider terms that are of potential relevancy to the end user. Typically the term will be a Noun or Noun Phrase.

Spelling:

The words used for the English Term used should conform to the Oxford English Dictionary (OED). However, commonly used English terms or spellings that are nation-specific could be considered for inclusion as additional RWOs synonyms (e.g. the American spelling of **Harbor**).

Non English Terms:

If a RWO from another language needs to be translated to the English equivalent. In a rare case when a non-English RWO cannot be translated into English, consideration will be given to using the term in the original language. Justification is required. The Description (E.1.12) should be used to record the origin language and justification.

Uniqueness:

The English Term shall be unique for each entry in the DGRWI. Effort should be made to distinguish between similar but semantically different objects. The only exception for using a non-unique term is the re-introduction of a former, previously retired term with new encoding. The Description (E.1.12) should be used to record justification.

Context Disambiguation:

If two or more distinct RWOs result in the same English term (i.e. a single English word with two different meanings) can have different meanings (homonyms), e.g. Bank, the English Term in DGRWI should be disambiguated using parentheses to distinguish subject or context, e.g. **Bank (financial)** and **Bank (riverside)**

Territorial Disambiguation:

If an object name can have semantically different meanings in different English speaking territories, e.g. Subway, and cannot be distinguished by context, the English Term in DGRWI should be disambiguated using parentheses to distinguish appropriate territory e.g. **Subway (USA)**. Common territories could include GBR, USA, Commonwealth, North America.

E.1.7 Profiles [Mandatory]

Information allowing the identification of the profile usage of the concept in DGIF applications/specifications. The impact of any change proposed on this concept can be identified and communicated.

E.1.8 Significance [Mandatory]

A RWO Index item will have an indication of the significance of the object to the user. Each RWO item will have one of the following values:

- **Modelling** = the default value.
- **Basic** = A RWO name that is a core, commonly used term used in the military environment. This includes terms that are both formal military terminology, as well as everyday colloquial terms, e.g. **Demilitarized Zone, Hospital, Divided Highway**.
- **Extended** = A RWO term that has been derived from Feature and Attribute values. These often result in non-standard English terms that, though make sense and describe an object, are not in everyday usage. **Alaska Pollock Fishing Area, Sail Maker at Small Craft Facility, Human Tissue Repository Facility**.

E.1.9 Source [Optional]

Indicates the source for the RWO which supports the DGRWI management process. This may include a reference to a formal glossary of terminology.

E.1.10 OCL Statement [Mandatory]

The pure OCL statement for the constraint. This contains the OCL encoding of the tuple(s) that defines the Object. The OCL statement and the OCL comment are normative and have to be copied in the constraint of the RWO class (E.1.3). The OCL statement starts with the term "entity" followed by consecutive numbers of elements used for the representation.

inv: entity1.ditchFunction.valueOrReason.value=Ditch_ditchFunction::irrigation

See Annex C for detailed information about correct OCL syntax

E.1.11 OCL Comment [Mandatory]

A human readable definition of the OCL statement. This contains the OCL representation of the tuple(s) that defines the Object. The OCL statement and the OCL comment are normative and have to be copied in the constraint of the RWO class.

/ Ditch has attribute ditchFunction with value irrigation */*

See Annex C for detailed information about correct syntax.

Care should be taken to ensure the logic of this statement when read as English is consistent with the concept (care with 'and', 'or' statements; introduce 'where' and 'if' etc. as required). Avoid logic ambiguity.

E.1.12 Description [Conditional]

A general supporting comment about the RWO:

A RWO may have additional information attached to better describe the concept if this is unclear from the term. This could include a description of the disambiguation.

E.1.13 Status [Mandatory]

The Concept's status as defined in ISO 19135. Only Concepts designated as "valid" are published for use. Other states depend on the management process described in 'HBK-WD-13-044-Change Management for DGIF (CM-DGIF)'

E.1.14 Date Accepted [Mandatory]

The date the Concept was accepted for DGRWI after an official voting process.

E.1.15 Date Retired [Conditional]

The date the Concept was retired for DGRWI after an official voting process.

E.2 Representation Stereotype

The Representation link is created from the RWO to all the relevant elements (featureType or type) that are required/referenced in the RWO.

E.2.1 Target Role (Mandatory)

The target role of the element should be named with "entityX", where "X" is a consecutive number for the RWO.

For Example the first (or only) Target Role will be set as "entity1". All subsequent Representations (i.e. where more than one Feature Type or Type is used) should use "entity2", "entity3", etc.

These entries are directly referenced in OCL statements.

Annex F - DGRWI Queries

A series of SQL queries have been developed to help DCE users review and maintain the DGRWI content. This list will continue to evolve.

Document EDT-WD-15-032-Queries Documentation provides details on all DCE Queries and how to utilise them.

F.1 Identify Next available RWO 531Code

Purpose:

A query to give an overview on used codes (531code) for realWorldObjects in the case one has to create a new realWorldObject and needs a free code.

SQL Statement:

```
SELECT
t_object.ea_guid AS CLASSGUID,
t_object.Object_Type AS CLASSTYPE,
t_objectproperties.Value AS C531Code, t_object.Name AS AlphaCode
FROM t_object LEFT JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE t_object.Stereotype='realWorldObject' AND
t_objectproperties.Property='531Code'
ORDER BY t_objectproperties.Value DESC;
```

Output:

All realWorldObjects with 531Code and alphaCode sorted by decreasing 531Code.

F.2 QA Check Alphacode Entries

Purpose:

To visually check ROI Alphacode entry for consistency with entry rules

SQL Statement:

```
SELECT t_object.CreatedDate, t_object.ModifiedDate, t_object.Author, t_object.Name,
t_object.Alias, t_object.Scope, t_object.NType, t_object.Tagged, t_object.Stereotype
FROM t_object
WHERE (t_object.Stereotype)='realWorldObject'
ORDER BY t_object.CreatedDate, t_object.Name;
```

Checks to make:

Check for Alphacode errors against Entry Rules above, specifically;

- Where an Alphacode is null
- Where an Alphacode doesn't begin with a Capital Letter
- Where an Alphacode contains spaces
- Where an Alphacode doesn't begin with a Letter or Number
- Where an Alphacode contains diacritics

- Where an Alphacode is not unique amongst all Alphacodes
- Check Spellings!

F.3 QA Check 531 Codes

Purpose:

To visually check RWO Alias/531 Codes for consistency with entry rules

SQL Statement:

```
SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_object.Alias AS AliasCode,
t_objectproperties.Property, t_objectproperties.Value AS Code
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE ((t_object.Stereotype)='realWorldObject') AND
((t_objectproperties.Property)='531Code')
ORDER BY t_object.CreatedDate, t_object.Name;
```

Checks to Make:

Check for Alias and 531 Code errors against Entry Rules above, specifically;

- Where an Alias doesn't equal the 531 code
- Where an Alias is null
- Where an Alias is not unique amongst all Aliases
- Where a 531 Code is null
- Where a 531 code does not equal 9 characters
- Where a 531 code does not begin with RWO_
- Where a 531 code is not unique amongst all 531 code

F.4 QA Check English Term

Purpose:

To visually check the RWO English Term for consistency with entry rules

SQL Statement:

```
SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_objectproperties.Property,
t_objectproperties.Value AS EnglishName
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE ((t_object.Stereotype)='realWorldObject') AND
((t_objectproperties.Property)='englishTerm')
ORDER BY t_object.CreatedDate, t_object.Name;
```

Check to make:

Check for English Term errors against Entry Rules above, specifically;

- Where English Term is Null

- Where English Term is not Unique amongst other English Terms
- Where English Term and Alphacode are inconsistent
- Spellings
- Identify any synonym recommendations

F.5 QA Check Source

Purpose:

To visually check RWO source for consistency with entry rules

SQL Statement:

```
SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_objectproperties.Value AS Source
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE (((t_objectproperties.Property)='source') AND
((t_object.Stereotype)='realWorldObject'))
ORDER BY t_object.CreatedDate, t_object.Name;
```

Checks to make:

Check for Source errors or consistency, specifically;

- Where Source is Null

F.6 QA Check Description

Purpose:

To visually check the RWO Description for consistency with entry rules

SQL Statement:

```
SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_objectproperties.Property,
t_objectproperties.Value AS description
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE ((t_object.Stereotype)='realWorldObject') AND
((t_objectproperties.Property)='description')
ORDER BY t_object.CreatedDate, t_object.Name;
```

Checks to make:

Check for Description errors against Entry Rules.

F.7 QA Check Status

Purpose:

To visually check the RWO Status for consistency with entry rules

SQL Statement:

```
SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_object.Stereotype,
t_objectproperties.Property, t_objectproperties.Value
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE (((t_object.Stereotype)='realWorldObject') AND
((t_objectproperties.Property)='Status') AND ((t_objectproperties.Value)='Superseded'))
OR (((t_objectproperties.Value)='Retired'))
ORDER BY t_object.CreatedDate, t_object.Name;
```

Checks to make:

Check for Status errors against Entry Rules;

- Review entries with status retired
- Review entries with status Superseded

F.8 QA Check OCL**Purpose:**

To visually check the OCL Statement and OCL Comment for consistency with each other and entry rules

SQL Statement:

```
SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_object.Note,
t_objectproperties.Property, t_objectproperties.Notes AS OCLStatementOCLComment]
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE (t_object.Stereotype ='realWorldObject') AND
(((t_objectproperties.Property)='oCLComment') OR
((t_objectproperties.Property)='oCLStatement'))
ORDER BY t_object.CreatedDate, t_object.Name, t_objectproperties.Property;
```

Checks to Make:

Check for OCL Statement and OCL Comment errors against Entry Rules above, specifically;

- Where OCL Statement and OCL Comment are null
- Where OCL Statement and OCL Comment are inconsistent

F.9 QA Check OCL Comment**Purpose:**

To visually check OCL Comment for consistency with entry rules

SQL Statement:

```

SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_object.Note, t_objectproperties.Notes
AS OCLComment
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE (t_object.Stereotype='realWorldObject') AND
((t_objectproperties.Property)='oCLComment')
ORDER BY t_object.CreatedDate, t_object.Name, t_objectproperties.Property;

```

Checks to Make:

Format errors in OCL Comment

F.10 QA Check OCL Statement

Query:

To visually check OCL Statement for consistency with entry rules

SQL Statement:

```

SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_object.Note, t_objectproperties.Notes
AS OCLStatement
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE (t_object.Stereotype='realWorldObject') AND
((t_objectproperties.Property)='oCLStatement')
ORDER BY t_object.CreatedDate, t_object.Name, t_objectproperties.Property;

```

Checks to Make

Format errors in OCL Statement

F.11 QA Check Significance

Purpose:

To visually check RWO Significance for consistency with entry rules

SQL Statement:

```

SELECT t_object.CreatedDate AS CreatedDate, t_object.ModifiedDate AS ModifiedDate,
t_object.Author, t_object.Status, t_object.Name, t_objectproperties.Property,
t_objectproperties.Value AS significance
FROM t_object INNER JOIN t_objectproperties ON t_object.Object_ID =
t_objectproperties.Object_ID
WHERE ((t_object.Stereotype)='realWorldObject') AND
((t_objectproperties.Property)='significance')
ORDER BY t_object.CreatedDate, t_object.Name;

```

Checks to Make

Check for Significance errors or possible re-categorising against Entry Rules, specifically;

- Where Significance is Null
- Where Significance = Basic.
- Where Significance = Modelling.
- Where Significance = Extended

F.12 List of all Real World Objects linked to a given Entity.

Purpose:

A query to create a list of all Real World Objects linked to a given Entity.

SQL Statement:

```
SELECT
t_object_1.Object_Type AS CLASSTYPE,
t_object_1.ea_guid AS CLASSGUID,
t_object_1.Name,
t_objectproperties_1.Value AS C531,
t_objectproperties_2.Notes AS oclStatement,
t_objectproperties.Notes AS OCLComment,
t_objectproperties_3.Value AS EnglishTerm
FROM
((((t_object INNER JOIN t_connector ON t_object.Object_ID =
t_connector.End_Object_ID) INNER JOIN t_object AS t_object_1 ON
t_connector.Start_Object_ID = t_object_1.Object_ID) INNER JOIN t_objectproperties AS
t_objectproperties_1 ON t_object_1.Object_ID = t_objectproperties_1.Object_ID) INNER
JOIN t_objectproperties AS t_objectproperties_2 ON t_object_1.Object_ID =
t_objectproperties_2.Object_ID) INNER JOIN t_objectproperties ON t_object_1.Object_ID
= t_objectproperties.Object_ID) INNER JOIN t_objectproperties AS t_objectproperties_3
ON t_object_1.Object_ID = t_objectproperties_3.Object_ID
WHERE
t_object.Name='<Search Term>' AND t_object.Stereotype='featureType' AND
t_object_1.Stereotype='realWorldObject' AND t_objectproperties_1.Property='531Code'
AND t_objectproperties_2.Property='oCLStatement' AND
t_objectproperties.Property='oCLComment' AND
t_objectproperties_3.Property='englishTerm';
```