IFC Tunnel - Conceptual Model Report

Annex I – Reading guide

InfraRoom

Project:	IFC Tunnel
Work Package:	WP3
Author(s):	IFC Tunnel
Date:	2022.06.28
Version:	V1.0
Status:	FINAL

CONTENT

1 Annex I – Reading guide

1.1 Introduction

This document provides a guide for how to read the IFC Tunnel Conceptual Model Report documents.

As a language for the conceptual model, UML (https://www.uml.org/) is used for the entire infrastructure domain. To achieve uniformity within the Infra domain and to provide a usable input for IFC schema generation, certain guidelines regarding UML modeling were agreed. These guidelines are explained in this guide and this will help the reader of the conceptual model report to better understand its content.

1.2 Document structure

The conceptual model and the report are sub-divided into topics. These topics are in UML represented as packages. Packages may contain other (smaller) packages, i.e. sub topics, and each package also contains the concepts belonging to that topic.

Packages are illustrated in UML as in the figure below:

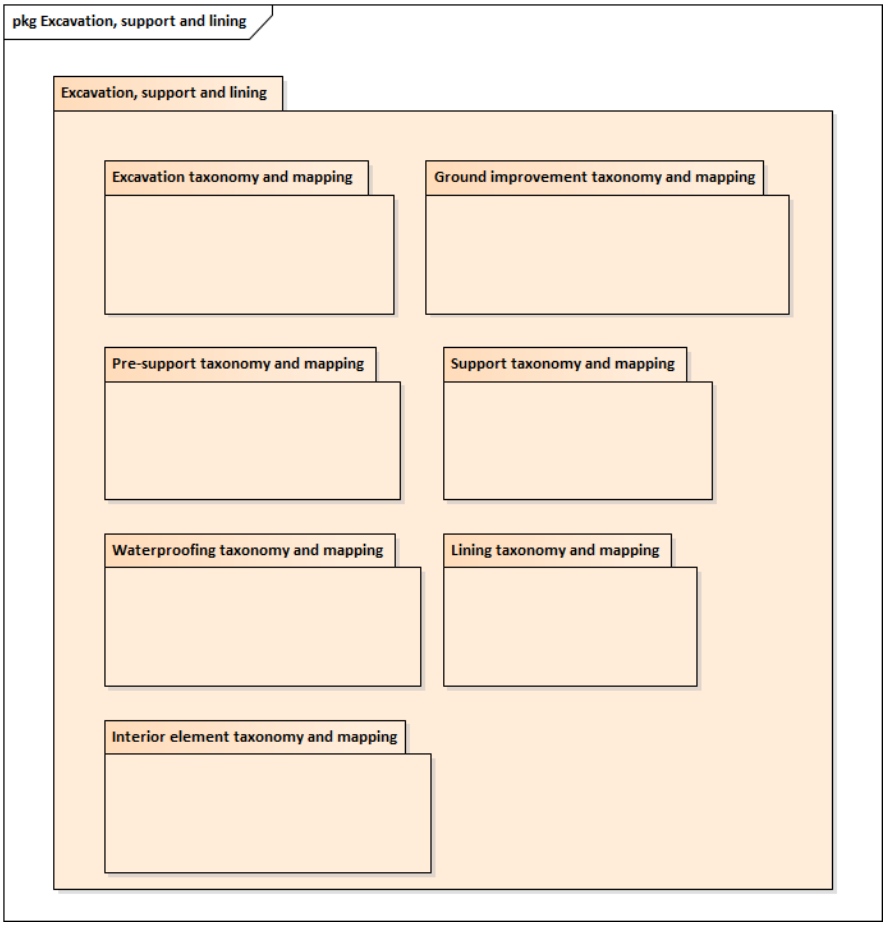


Figure 1 - UML package depiction

The UML package may be depicted with or without its content. This package contains seven sub packages and these sub packages contains a number of concepts.

In the report, each UML package has its own chapter, and the chapter structure directly corresponds to the package structure in the UML model. For each package, there is typically one or more UML class diagrams illustrating the content of the package.

1.3 Concepts

As input for the conceptual model, the concept definitions from the requirements analysis phase and the taxonomies produced by the domain expert teams (Geotechnics, Excavation, support and lining and Systems) were used. Depending on the nature of a concept it may become different things in the UML model.

For those concepts that are considered as being entities in IFC, UML Classes (without stereotype) are created. Classes are depicted in diagrams as illustrated in the figure below.

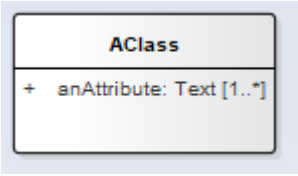


Figure 2 - A depiction of a class

Each class shall have name and a definition and may have properties (or attributes) which are listed in the compartment below the class name. The class name is shown in the upper compartment (AClass in the above figure). Attributes (properties for the class) are listed in the compartment below the class name and are described using the following form:

Attribute: Type [m..n]

Where “Attribute” is the attribute name, Type is the type of the attribute and within brackets “[]” any cardinality restrictions is shown (in UML named multiplicity). If the cardinality restriction is omitted, a multiplicity of exactly 1 is assumed. In the figure above, we have a class named “AClass” with one property with the name “anAttribute” which has the type (is represented as) “Text” and that may occur one to many times for each instance. The type of an attribute may be a primitive type, an enumeration or another class.

Classes may be arranged as a taxonomy (see chapter 1.4), allowing more specific concepts to be specializations (sub classes) of more generic concepts.

Besides occurring in a class diagram, a class is documented according to the following structure in the report:

1.3.1 Class: IfcXxx

Definition of IfcXxx.

Status: Proposed

Package: <Package name>

Class Properties			
Status	Proposed	**Is Abstract**	True or False
Property sets	Lists the property sets assigned to this class		

Inheritance Statement		
Subtype Of	If any, link to the more general class in the taxonomy	
Subtypes	**EXISTING**	**PROPOSED**
	If any, link to the already existing more specific classes in the taxonomy	If any, link to the proposed more specific classes in the taxonomy

Class Attributes (lists the attributes for the class if any)

Name	Type	Multiplicity	Definition
Attribute name	Attribute type	E.g. [0..1]	Definition of the attribute

1.3.2 Concepts as predefined types

In IFC, the notion of predefined types is used as means to represent lower level concepts as types of IFC entities. These predefined types are represented as an enumeration attribute with the name "PredefinedType". One example is the IFC entity IfcBeam that has the attribute PredefinedType of type IfcBeamTypeEnum which enumerates the occurring types, e.g. BEAM, JOIST, GIRDER_SEGMENT, CORNICE etc.

In the UML modelling for IFC Road we may choose to represent IFC Road concepts as proposed new IFC entities or as predefined types for existing or proposed new entities. Proposed new entities are represented as "normal" classes according to the previous chapter.

A concept which is proposed to be represented as a predefined type is modelled as a class with stereotype <<PredefinedType>>. The stereotype for a class is shown with the stereotype name at the top of the class box.

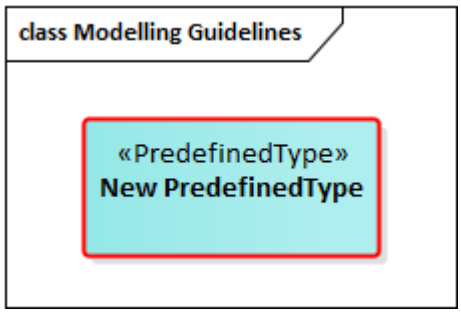


Figure 3 - A class defined as a predefined type

To group together all the predefined types belonging to a single parent IFC Entity, a class stereotyped as <<PTContainer>> is used according to the following example:

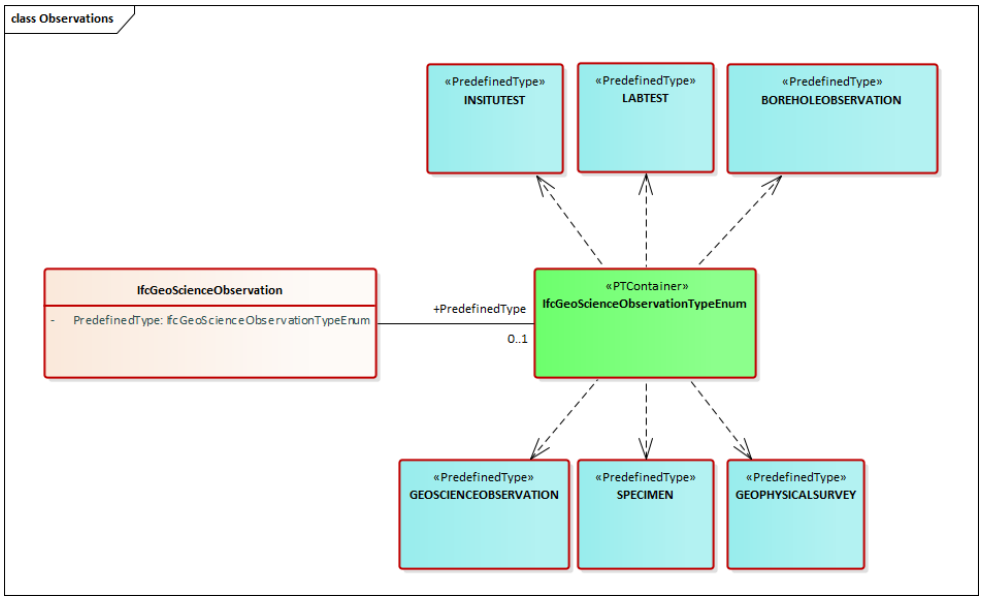


Figure 4 - Predefined types and container for predefined types

In the above example, the IFC Entity IfcGeoScienceObservation has predefined types defined. Each predefined type is represented as a <<PredefinedType>> class and all predefined types are associated using a dependency relationship (the dashed arrows) to the container (<<PTContainer>>). The container is associated to the IFC entity with both an association and an attribute named PredefinedType.

A predefined type container is documented according to the following structure in the report:

1.3.3 PDT Container: Name of the container (e.g. IfcGeoScienceObservationTypeEnum)

Definition…

Status: **Proposed**

Package: E.g. Earthworks Cut Elements

Container Properties			
Parent Entity	Link to the owning IFC Entity (e.g. IfcGeoScienceObservation)	**Stereotype**	«PTContainer»
Contains	**EXISTING**	**PROPOSED**	
	Links to existing predefined types (if any)	Links to proposed predefined types (if any) E.g. IfcGeoScienceObservationTypeEnum.INSITUTEST IfcGeoScienceObservationTypeEnum.LABTEST Etc…	

A predefined type is documented according to the following structure in the report:

1.3.4 Predefined Type: Name of the predefined type (e.g. LABTEST)

Full Identifier: The fully qualified identifier (e.g. **IfcGeoScienceObservationTypeEnum.LABTEST)**

Definition…

Status: **Proposed**

Package: E.g. Earthworks Cut Elements

Predefined Type Properties			
Predefined Type Container	Link to the container for the predefined type E.g. IfcGeoScienceObservationTypeEnum	**Parent Entity**	Link to the Entity owning the container E.g. IfcGeoScienceObservation
Stereotype	«PredefinedType»		
Property sets	Links to assigned property sets if any		

1.4 Taxonomic structure

The concepts from the requirements analysis as well as the classes in a UML model are typically organized in a taxonomic structure where classes are organized in generalization/specialization relationships.

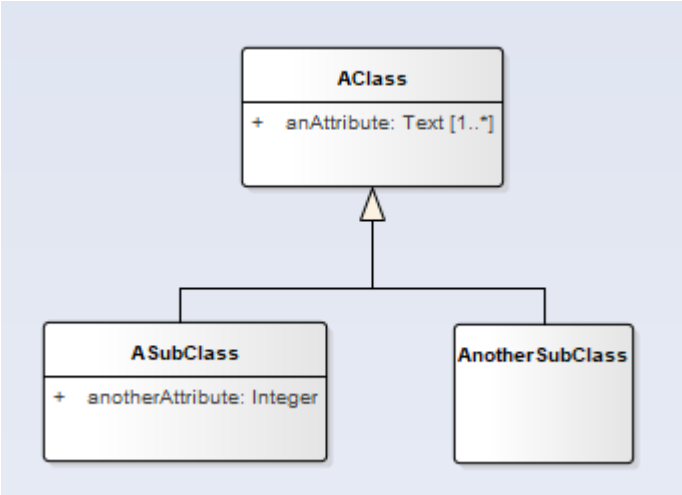


Figure 5 - A class taxonomy in UML

In the figure above, the classes ASubClass and AnotherSubClass are specializations of the more general class AClass. This relationship is depicted with the kind of arrow going from the specialization (sub class) to the generalization (super class). According to set theory, these classes may be viewed as sets where an instance belonging to the set ASubClass also belongs to the set AClass.

In some diagrams, where the super class of a class is not present, the name of the super class is written in the sub class box as in the figure below.

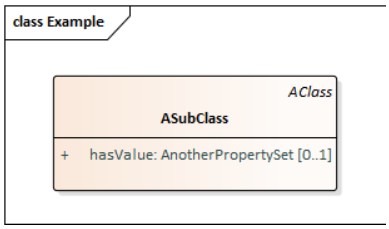


Figure 6 - Class with its ancestor (super class) not present in the same diagram

1.5 Property sets and Quantity sets

Property sets represent a flexible way of attaching properties that are not pre-defined in the IFC schema to entities; in a similar manner, Quantity sets provide a flexible way of attaching quantity data. The IFC Tunnel project still has to define tunnel specific property sets and quantity sets in addition to the already existing ones. A property set or quantity set is applicable for one or more IFC entities and/or predefined types. In UML a representation is needed to represent the property set itself as well as its applicability.

In the example below, we see the IFC Road property set Pset_PavementMillingCommon (each property set name shall be prefixed with “Pset_”). The property set is modelled as a class with stereotype <<PropertySet>> and is rendered with a separate color, to clearly separate it from ordinary classes. The property set has two properties, Width and Depth, that are described just as any properties for any class. Lastly, the property set is assigned to a class, in this case the predefined type PAVEMENTMILLING. This assignment is done via a Realization relationship (dashed arrow) from the class to the property set.

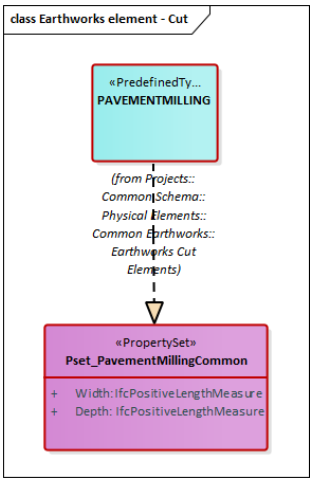


Figure 7 - Property set assignment example

In the example below, the quantity set Qto_MemberBaseQuantities (each quantity set name shall be prefixed with "Qto_") and its applicability to the entity IfcMember is illustrated. The quantity set is modelled as a class with stereotype <<QuantitySet>> and is rendered with a separate color, to clearly separate it from ordinary classes. The quantity set has nine properties that are described just as any properties for any class. Lastly, the quantity set is assigned to a class, in this case the class IfcMember. This assignment is done via a Realization relationship (dashed arrow) from the class to the quantity set.

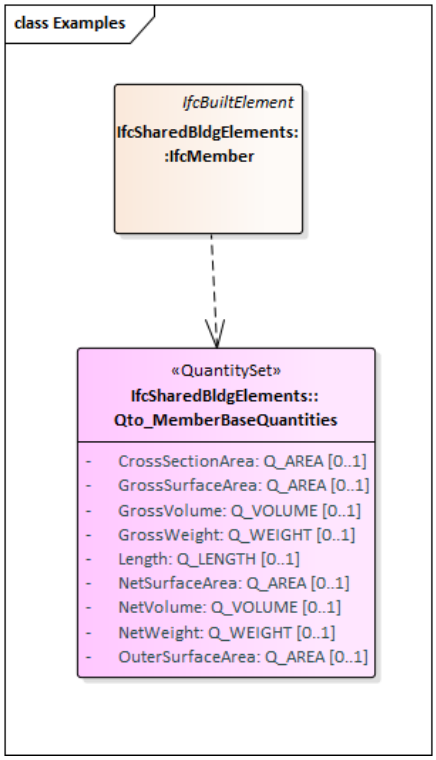


Figure 8 - Quantity set assignment example

In the report, a property set is documented according to the following structure (similar for a quantity set):

1.5.1 Property Set: Name e.g. Pset_PavementMillingCommon

Status: **Proposed**

Set Properties			
Applicable Entities	Link to the classes to which the property set is assigned, e.g. IfcEarthworksCutTypeEnum.PAVEMENTMILLING	**stereotype**	«PropertySet»

Properties (lists the properties in the property set)

Name	Type	Multiplicity	Definition
Width	IfcPositiveLengthMeasure		The nominal width of the milling. The size information is provided in addition to the shape representation and the geometric parameters used within. In cases of inconsistency between the geometric parameters and the size properties, provided in the attached property set, the geometric parameters take precedence.
Depth	IfcPositiveLengthMeasure		The nominal milling depth. The size information is provided in addition to the shape representation and the geometric parameters used within. In cases of inconsistency between the geometric parameters and the size properties, provided in the attached property set, the geometric parameters take precedence.

The classes to which the property set was assigned will have the corresponding link to the property set in their “Property sets” documentation box.

1.6 Enumerated types for properties in property sets

In certain cases, the type of a property in a property set may be an enumeration of values. These are modelled as UML enumerations with the stereotype <<PEnumType>>.

The property set below has the property Basis of type PEnum_UncertaintyBasis which is an enumeration.

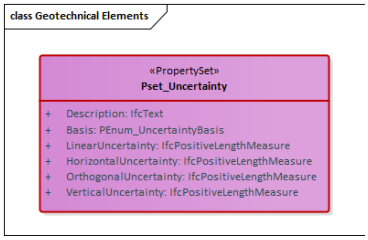


Figure 9 - Property set with property that has enumerated type

The enumerated type is modelled as in the figure below.

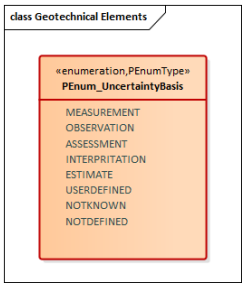


Figure 10 – A PEnumType

Each of the possible values is listed in the enumeration box. Such an enumeration is documented in the report in the following way:

1.6.1 Enumeration: Name e.g. PEnum_UncertaintyBasis

The definition.

Status: Proposed

Package: Geotechnical Elements

Enumerators

Name	Definition
MEASUREMENT	A definition
OBSERVATION	A definition
ASSESSMENT	A definition
INTERPRITATION	A definition
ESTIMATE	A definition
USERDEFINED	A definition
NOTKNOWN	A definition
NOTDEFINED	A definition

1.7 Virtual entities

The concepts from the requirements analysis and domain taxonomies shall be mapped to existing or proposed IFC Entities or predefined types that fulfills the requirements of the concepts.

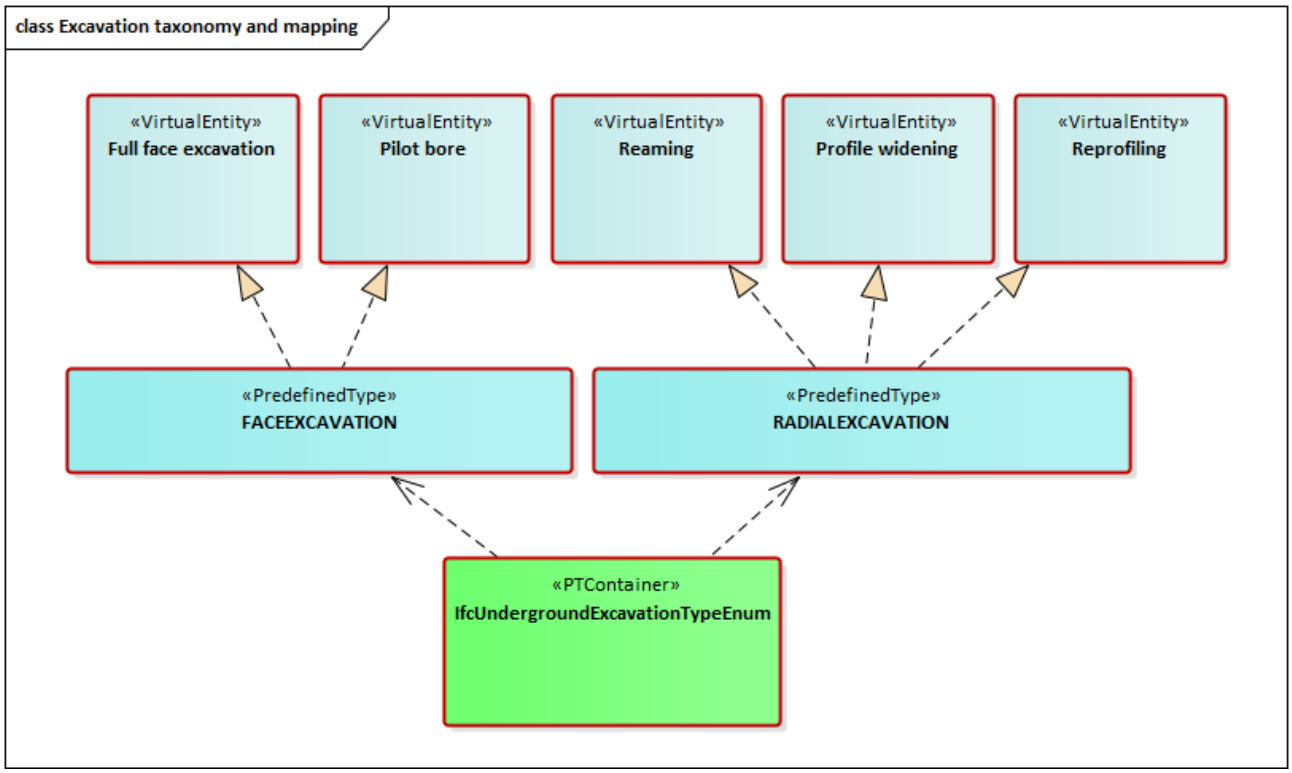


Figure 11 - Virtual entity example

In the example above, the concepts “Full face excavation”, “Pilot bore”, “Reaming”, “Profile widening” and “Reprofiling” are modeled as <<VirtualEntity>>. The virtual entities are rendered using a separate color and are linked to the entity or predefined type that realizes it using a Realization relationship (dashed arrow) going from realizing class to the realized virtual entity.

A virtual entity is documented in the following way in the report:

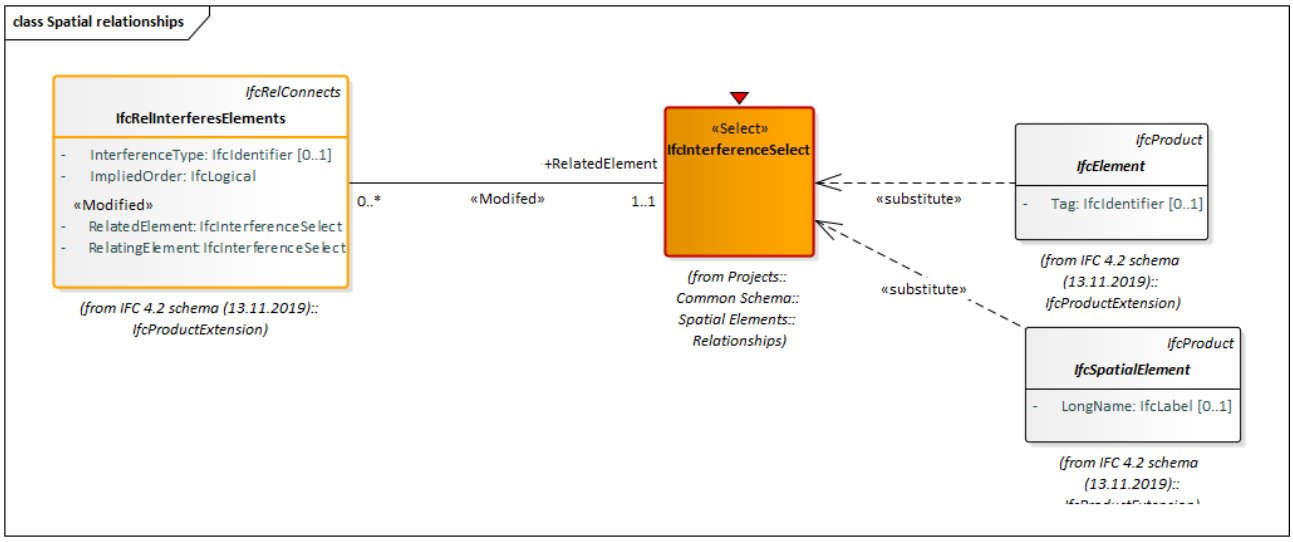
1.7.1 Virtual Entity: Name e.g. Full face excavation

Definition…

Entity Properties	
Realizing Parent	Links to the classes that realizes the virtual entity, e.g. IfcUndergroundExcavationTypeEnum.FACEEXCAVATION
Notes	Any notes attached to the virtual entity or realization, e.g. specific conditions for the realization.

1.8 Select types

A select type is a kind of substitution grouping, where any of the enumerated types within the select type may be used in an instance model. This is modelled using a class stereotyped with <<Select>> and the class name suffixed with “Select”. The types (e.g. classes) that may be selected are associated with the select type using dependency relationships (dashed arrow) stereotyped with <<substitute>> as shown in the figure below. In the example below, the relationships RelatedElement and RelatingElement may use either an IfcElement or an IfcSpatialElement as their targets since their type is IfcInterferenceSelect.



1.9 Other relationships between concepts

UML allows for different kinds of relationships between concepts. Besides the relationships already illustrated above, only one is actually used in the IFC Road conceptual model and this is a normal association.

This kind of relationship is illustrated with a line between the classes according to the figure below.

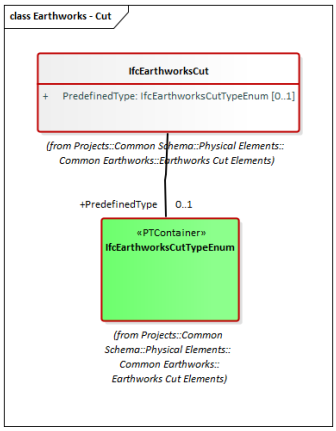


Figure 12 - Association example

The line between IfcEarthworksCut and IfcEarthworksCutEnum is a UML association and it is conceptually equivalent to the corresponding attribute inside the IfcEarthworksCut class box. However, it is good practice to show these associations in the diagram when both participating classes are present. The “PredefinedType” and “0..1” texts next to the lower end of the association specifies a role name and the multiplicity for that role. This is standard UML notation and indicates the role of a class in an association.

1.10 Color coding

To graphically separate different types of concepts and different status for concepts, the following coloring is used.

- The state is illustrated using different borders.
  - **Implemented**: applies to elements that have gone through public review and contained in a full standard. This is most relevant to the IFC 4x2 definition currently used within the model.
  - **Proposed**: applies to new elements added to the model.

- **Proposed Modification**: applies to existing elements that need to be changed (existing elements are those that have reached approved, Candidate or Implemented status within the model.
- **Deprecated**: applies to element that have previously been implemented but are now marked for removal from the standard
- **Approved**: applies to elements that have gone through expert panel review.
- **Candidate**: applies to elements that are included in a candidate standard release.

- The type (Entity, PredefinedType, PropertySet etc) is illustrated using different fill colors

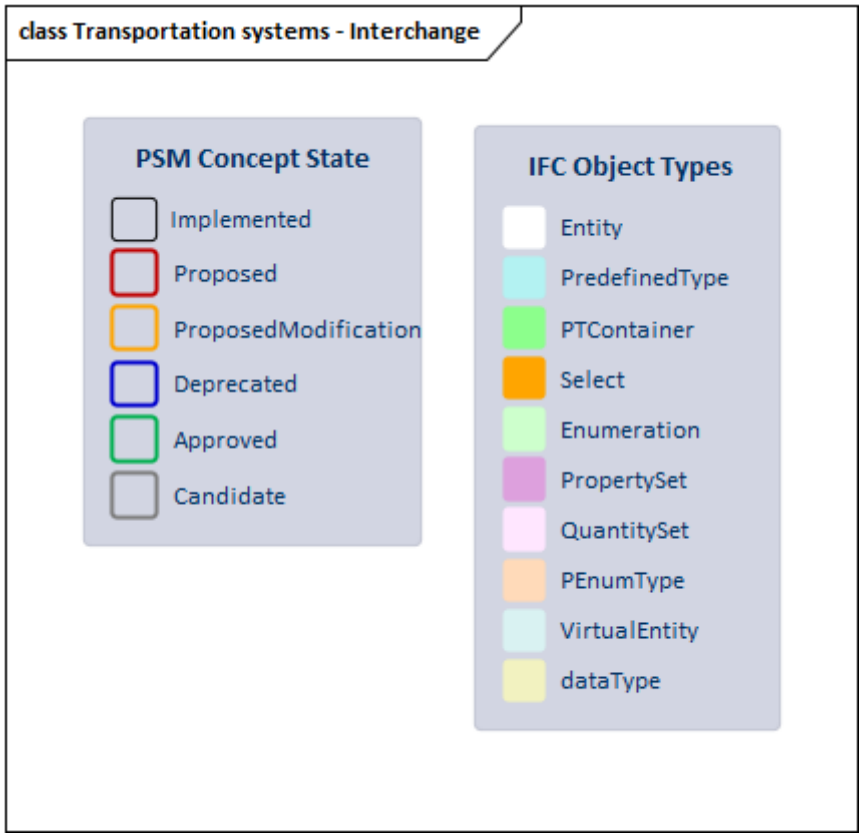


Figure 13 - Color coding

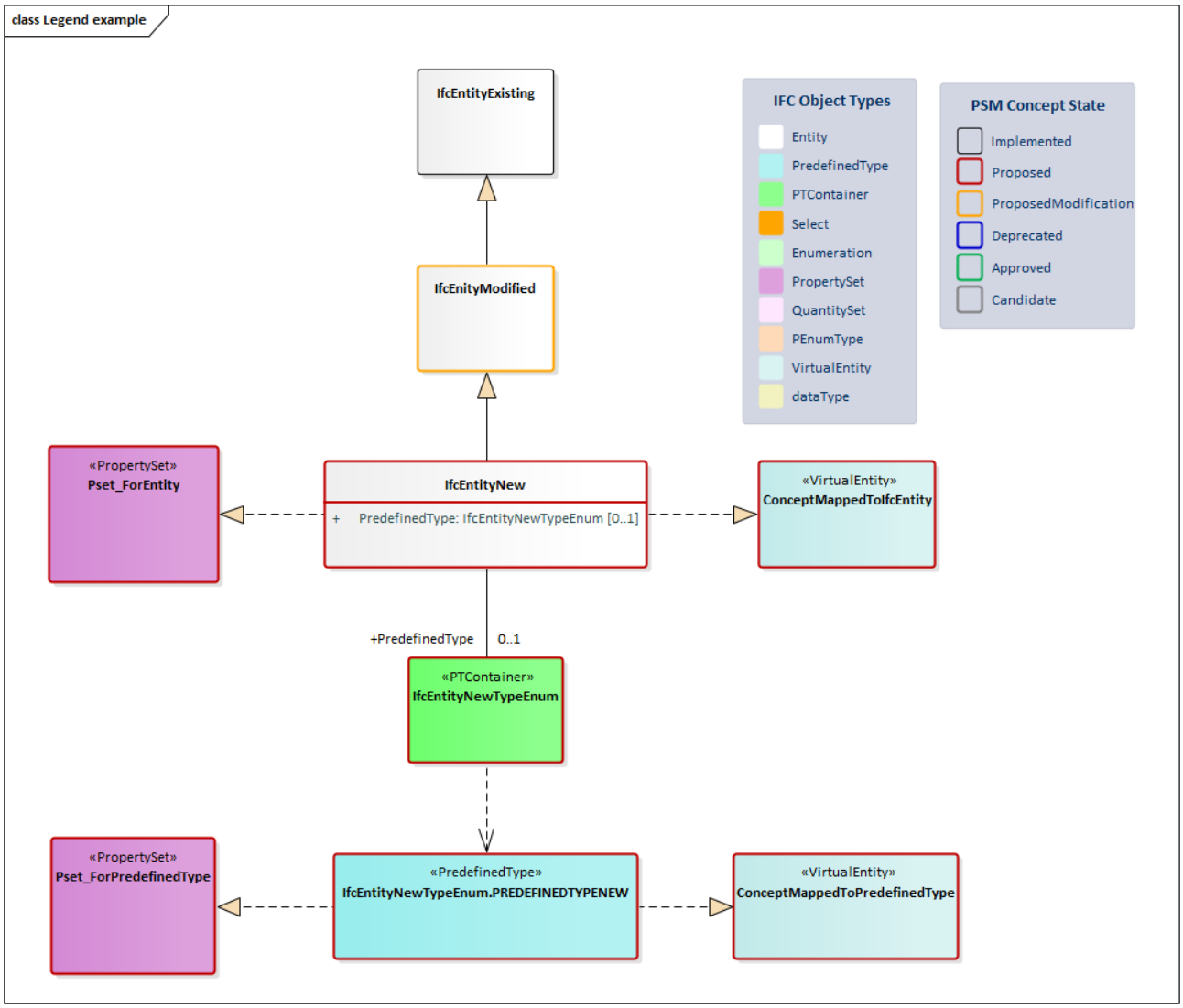


Figure 14 - Example color coding