

Part 1 - The UML Model Report

Introduction to the IFC Harmonised Schema Extensions

Project/Publisher: Common Schema / Infrastructure Room

Work Package: Common Schema - WP2 – Harmonisation & Development

Date: 24/04/2020

Version: **V04** – FINAL, PUBLISHED

Document Information

Document ID	Title	Created By	Created
IR-CS-WP2	bSI UML Model Report – Part 1	JP	2020-01-13

Revision History

Version	Status	Date	Notes
V01	DRAFT	2020-01-13	First draft – Part of Harmonised Document
V02	DRAFT	2020-01-31	Second draft – Part of Harmonised Document
V03	DRAFT	2020-02-03	Final draft – Part of Harmonised Document
V04	FINAL	2020-02-04	Final – Part of Harmonised Document
V05	PUBLISHED	2020-04-24	Corrections and appendices for publication

Author List

IFC Common Schema¹

Jim Plume	buildingSMART Australasia	Matthias Weise	AEC3
Jon Mirtschin	GeometryGym		

IFC Ports & Waterways

Prof. Haijiang Li	Cardiff University	Song Liu	CCCC
Alex Bradley	Cardiff University	Honglei Qin	CCCC
Nicholas Nisbet;	AEC3 UK	Xi Wen	CCCC
Mike Ramsay	Royal Haskoning DHV	Veronica Ruby-Lewis	Waldeck Consulting
Daniel Peel	Royal Haskoning DHV	Kyle Moss;	Waldeck Consulting
Maik Weidt	WSV Germany	Michael Kluge	Planen Bauen 4.0
Julia Wissel	WSV Germany		

IFC Rail²

Evandro Alfieri	Engisis	Chi Zhang	Applitec
Claude Marschal	R+P AG	Florian Hulin	SNCP Réseau
Matthiew Perin	Railenium	Thomas Liebich	AEC3
Sebastian Esser	TUM		

IFC Road³

Hyouonseok Moon	KICT	Karin Anderson	Swedish Transport Agency
André Borrmann	TUM	Juha Hyvärinen	Jhy OY
Štefan Jaud	TUM	Lars Wikström	Triona
Sergej Muhič	Siemens & AEC3	Johnny Jensen	Trimble
Antonio Marquez	Apogea	Joaquim Moya	Apogea
Jaeyoung Shin	KICT	FeiFei Zhao	CRBIM
Jisun Won	KICT		

1. Complete contributor list for Common Schema can be found in Appendix B
2. Complete contributor list for IfcRail can be found in Appendix C.
3. Complete contributor list for IfcRoad can be found in Appendix D.

Table of Contents

Foreword	4
Report Structure	4
Executive Summary	5
1 Background and Context	6
1.1 International Consensus	7
1.1.1 IFC Road Project	7
1.1.3 IFC Ports & Waterways Project	8
1.1.4 IFC Common Schema Project	8
2 Harmonisation Process	8
3 Next Steps	11
4 Scope of the Harmonised Schema	12
4.1 Rail Project	12
4.2 IFC Road	14
4.3 Ports & Waterways Project	15
Appendix A – bSI UML Modelling Guidelines (DRAFT)	16
Appendix B – IFC Common Schema Contributor List	65
Appendix C – IFC Rail Contributor List	67
Appendix D – IFC Road Contributor List	69

Foreword

The purpose of this report is to describe the process followed in order to arrive at a harmonised IFC schema extension for infrastructure, building on previous work on Alignment, Overall Architecture and IFC Bridge (published by bSI as a Candidate Standard and known as IFC 4.2). The proposed extension schema is documented using UML language and diagrams and incorporates the development work completed over 2018-2019 for Rail (and its sub-domains), Road, Ports & Waterways (collectively referred to as the domain projects in this document) and Common Schema. The full report is in 5 Parts, the first being this present document that provides the background introduction to the full harmonisation process and separate parts for each of the three project domains plus a part covering all the shared infrastructure elements.

Note that the Common Schema project responded to the identified needs of the domain projects to develop proposed extensions for specific common areas (such as spatial structure, geotechnics, earthworks, etc.), but during the harmonisation process, many other shared elements were identified and collected together into the common schema package. For the purposes of this report, all those are collectively treated as shared infrastructure elements.

The complete 5-Part Report is submitted to the bSI Standards Committee Executive as documentation in support of the adoption of the of the harmonised IFC infrastructure schema extension as a Candidate Standard.

Report Structure

This document forms Part 1 of the Report and is intended to explain the background, context and overall process followed to deliver the harmonised schema. The other four Parts are auto-generated directly from the harmonised UML conceptual model that has been created collaboratively between the separate domain project teams (in a process that will be explained further in this document). Those Parts document all proposed new entities, predefined types, modified entities (from the IFC 4.2 base line) and any entities proposed to be deprecated as part of this harmonisation work.

Part 2 reports the UML conceptual model that deals with all those concepts and specifications that are shared across the separate project domains (referred to collectively as shared infrastructure elements).

Parts 3-5 report the UML conceptual models that deal with the separate project domains of Road, Rail and Ports & Waterways: it is important to understand that these are self-contained and complete reports for each of those domain projects and therefore may include common schema concepts that are relevant to that separate domain; that leads to a lot of duplication and repetition in those Parts. It was deemed appropriate to allow that duplication in order to maintain the integrity of those Parts, which stand as important deliverables for the Stakeholders for those domain projects.

Executive Summary

Throughout 2018/19 buildingSMART ran a suite of parallel projects to deliver proposals for IFC extensions for infrastructure each with a focus on a specific project domain. There was general recognition of the need for a formal collaborative harmonisation process following a Workshop convened by the Common Schema Project on 14-15 January 2019 in Helsinki. That Workshop included representatives from all the domain projects and marked a significant move in the establishment of a strong collaboration between the Railway Room and the Infrastructure Room. A Delivery Plan was developed and refined in conjunction with Harmonisation Workshops held in June, September, October and November 2019. A robust harmonisation process was agreed in the leadup to the Beijing Summit and implemented over the period between October 2019 to January 2020.

The harmonisation process was based on a centralised (cloud) environment that hosted a structured UML conceptual model based on a UML encoding of IFC 4.2, therefore including the alignment specification and the Bridge candidate standard. The software tool used was Enterprise Architect, which allowed the harmonised model to be organised into a hierarchy of packages representing each of the separate project domains (plus the baseline IFC 4.2 encoding and the Common Schema) at the top level, with sub-packages established to suit the needs of each project domain. Since all the conceptual models were held in a single place, inconsistencies and conflicts could be identified and resolved in a collaborative process.

The final unified UML conceptual model is documented in this report (generated directly from Enterprise Architect), but it also forms the basis for the delivery of the formal specification using ifcDoc. The complete harmonised UML conceptual model can be exported from Enterprise Architect and imported (with some minor tweaks) into ifcDoc and merged with the entire IFC specification.

A critical part of this process was the development of UML Modelling Guidelines (see Appendix A) that describe the way the UML model has been built, with the general rules and patterns that have been followed. This ensures consistency, facilitates the publication process and clarity of the UML diagrams presented in this report, and smooths the transfer process from the UML model (exported in XMI format) to the ifcDoc repository. It is important to note that while 95% of the guidelines are appropriate to UML as a general-purpose conceptual modelling language, there are parts that make use of constructs supported specifically by Enterprise Architect, employed here to facilitate the harmonisation process and conversion to an IFC specification held in the ifcDoc repository.

The harmonisation process has focussed on the proposed IFC entities (new and modified existing) and associated predefined types. Some domain Property Sets have been specified as part of the domain projects, but much of that remains as a work in progress. An important next step in harmonisation (deliberately postponed until 2020) is to identify and resolve conflicts and duplication of terms in those property sets.

The ifcDoc repository will be important as we move to the next stage in this work, taking the harmonised schema and validating it through a rigorous deployment and software implementation process based on identified test cases. That work is planned for 2020 and is the subject of a separate project proposal.

1 Background and Context

Throughout 2018/19 buildingSMART ran a suite of parallel projects to deliver proposals for IFC extensions for infrastructure. These were generally concerned with linear infrastructure and have been operating across differing time frames (Figure 2). It was recognised that these domain projects would be addressing several concepts that are common, so an overarching project was established called Common Schema, responsible to track those separate domain projects to identify and define common concepts (such as spatial structure, geotechnics and earthworks, utility networks, etc.) and to ensure a level of harmonisation and consistency in the development of the separate domain extensions.

Two of those projects (Road and Ports & Waterways), having completed a “requirements definition” stage, were developing conceptual models and draft schema of their proposed extensions during much of 2019. The Rail project prepared three major reports (Requirement Analysis, Conceptual Model and Data Requirements) as part of the Candidate Standard package delivered ahead of the Beijing Summit (28-31 October 2019). The Common Schema project delivered extension proposals for key areas including geotechnics, earthworks, spatial structure and kinematic envelopes. By the middle of 2019, it was recognised that there was an urgent need to harmonise the work across those domains, taking into consideration the work previously completed ahead of the IFC Bridge Candidate Standard.

There was unanimous agreement across all the infrastructure project domains that our collaborative goal would be to deliver a single harmonised IFC schema extension proposal that incorporated all the proposed extensions (road, rail, ports & waterways and common schema). It was agreed that this would be built upon the baseline defined by IFC 4.2 incorporating the alignment specification, the Bridge candidate standard and the schema extensions that came out of the work of the Overall Architecture project completed in 2017.

The context of this harmonisation work is the bSI standards development process shown in Figure 1. The proposed harmonised extensions have completed the development phase and, subject to a Standards Committee (SC) vote, will move to the approval phase as a unified candidate standard.

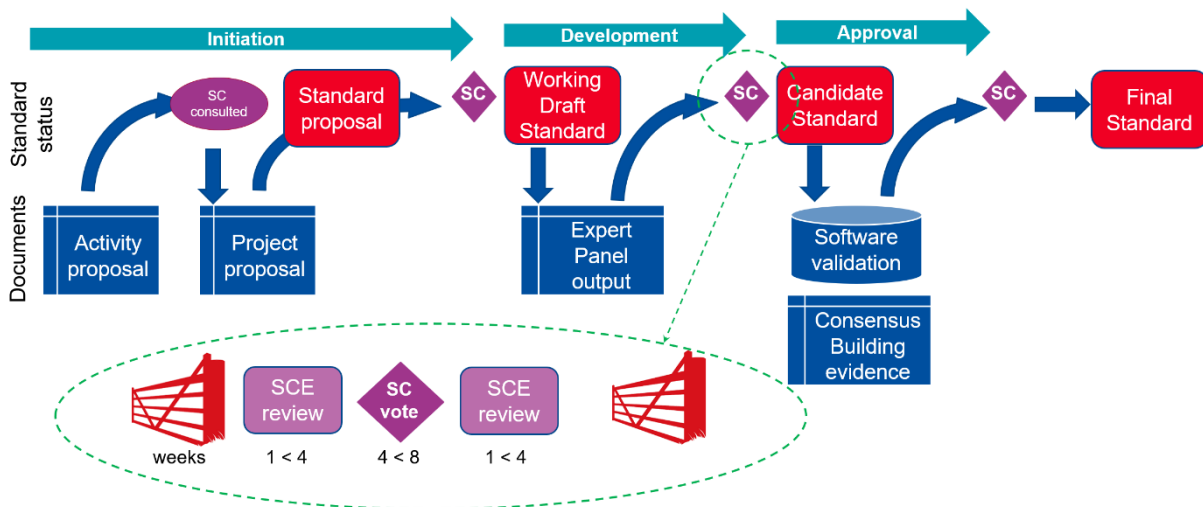


Figure 1 – bSI Standards development process

1.1 International Consensus

Evidence of international consultation through the participation Expert Panels is a critical part of the Development Phase of the process (Figure 1). Two of the domain projects involved in the harmonisation process (Bridge and Rail) have already been adopted as Candidate Standards. For the purpose of this document, it is necessary to report briefly on the Expert Panels held for Road, Ports & Waterways and Common Schema work.

1.1.1 IFC Road Project

The Road Project held a comprehensive set of Webinar-based Expert Panels to report progress to the international community and seek feedback.

Phase 1 - Requirements Analysis

- EP1, 2018.05.15 Use Cases and Road process diagram
(Registered 15; Attended 14)
- EP2, 2018.06.29 Taxonomy
(Registered 55; Attended 36)
- EP3, 2018.09.14 Draft IFC Road Requirement analysis Report
(Registered 67; Attended 42)
- EP4, 2018.11.15 Requirement Analysis report and Phase 2 Draft execution plan
(Registered 35; Attended 21)

Phase 2 – IFC Schema Extension and Deployment

- EP5, 2019.05.31 DRAFT Conceptual model
(64 comments received, + 10 input documents from Experts, Registered 308)
- EP6, 2019.08.29 DRAFT Schema extension
(36 comments received, Registered 248; Attended 125)
- EP7, 2019.10.24 DRAFT IFC Property definitions
(40 comments received+ 7 input documents, Registered 155)
- EP8, 2019.12.11 Implementation support and dissemination
(5 comments received, Registered 224)

Summits:

The project has been reporting the results and ongoing work at the summits in 2018, Paris and Tokyo and 2019, Dusseldorf and Beijing, and the final result including the harmonisation work was presented at the summit in Oslo March 2020.

Over 250 comments received and documents giving feedback from organisations worldwide. Feedback questionnaires were sent out following the webinars, these also gave an opportunity from the international community to comment on the work and to ensure it was relevant and in line with the needs of the industry. All webinars have been documented and recorded, anyone who registered has had access to the recording in case they were not able to attend the webinar itself.

1.1.3 IFC Ports & Waterways Project

The Ports & Waterways project held a mixture of Webinars and Summit presentations.

2018-03-27	buildingSMART Standards Summit Paris Ports and waterways session in the Infrastructure Room
2018-10-18	buildingSMART Standards Summit Tokyo Ports and waterways session in the Infrastructure Room
2018-12-12	“Ports and waterways requirements analysis” Online Webinar
2019-03-27	buildingSMART Standards Summit Dusseldorf Ports and waterways session in the Infrastructure Room
2019-11-26	(rescheduled from 2019-11-19) “Review of the draft IFC based conceptual model” Online Webinar

1.1.4 IFC Common Schema Project

The extension works packages delivered by the Common Schema Project held some dedicated Expert Panel Webinars, were reported at each of the bSI Summits in 2018 and 2019 and were also subject to review as part of the domain project Expert Panels. Specific Expert Panels were:

2018-09-13	Geotechnics Expert Panel (Webinar)
2018-09-27	IDBE Discussion on Geotechnics in Singapore (Inc BRGM, France)
2018-10-19	Tokyo Summit – IDBE Meeting
2018-12-05	Earthworks Expert Panel (Webinar)
2019-01: 22-24	3-day F2F Workshop on Geotechnics (hosted by BRGM, Paris)
2019-03-29	1-day F2F Workshop on Geotechnics (following Dusseldorf Summit)
2019-09-17	Earthworks, Geotechnics & Spatial Structure (on-line discussion), part of the Stockholm Harmonisation Workshop.
2019-10-21	Cross-Domain Telecon discussion on Spatial Structure

2 Harmonisation Process

Figure 2 shows the timeline and progress of the current work program including the anticipated work going forward, subject to project approval and funding. The key part of the diagram for this Report is the left part. The right part of the diagram shows the scope of work to be undertaken in 2020 and beyond and is included only to show the on-going context of the work.

The suite of domain projects that ran throughout 2019 are shown as bands in the diagram running up to the end of 2019, noting that Ports & Waterways will actually continue as a funded project up to May 2020, though the core schema development work was completed in step with the other projects. The Bridge project completed in early 2019.

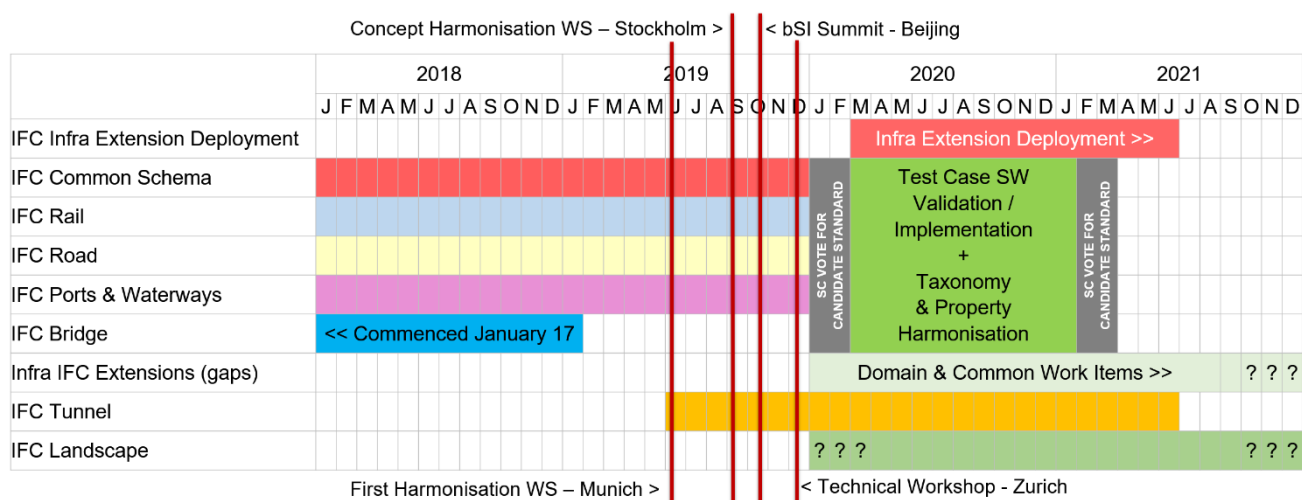


Figure 2 – Expected progress and completion of the current suite of projects

Throughout 2019, the Common Schema project hosted a series of workshops that sought to create a dialogue across all the parallel domain extension projects. The first, held on 13-14 January in Helsinki, brought together representatives from Road, Rail, Ports & Waterways and Bridge with the explicit goal to identify “user requirements for common concepts across those domains”. During the months following that workshop, there was a growing recognition that we needed a clear delivery plan that brought the separate domain project work together, particularly around the need to harmonise the IFC schema extension proposals. A formal Delivery Plan document was prepared and refined in parallel with a series of Harmonisation Workshops: 26-28 June in Munich, 16-17 September in Stockholm, 31 October in Beijing and a final Technical Harmonisation Workshop on 10-11 December in Zurich.

A significant outcome from this harmonisation effort has been an absolute commitment from the leaders of each of the current projects (Rail, Road, Bridge, Ports & Waterways and Common Schema) to work towards a single unified infrastructure schema extension proposal by the end of 2019. This is significant because it means that the three domain extension proposals (which have been developed as separate projects) and the Common Schema proposals are now fully harmonised at the schema level to support effective implementation in software and a consistent use of shared concepts. Since most infrastructure projects are inter-modal (having elements from across all the infrastructure domains as well as buildings), it is essential that the schemas be consistent at the conceptual level.

To achieve this harmonisation, a shared (cloud) modelling platform was set up and, following the Beijing Summit, a UML modelling team was established with (generally) 2 representatives from each domain charged with the responsibility to enter their own domain models and work with the other team members to resolve conflicts and identify/address commonalities. A combination of regular on-line meetings, a technical face-to-face meeting in Zurich in December 2019 and discussion threads on the BSI Forum were used to identify and resolve issues that arose.

The software tool used was Enterprise Architect, which allowed the harmonised model to be organised into a hierarchy of packages representing each of the separate project domains (plus the baseline IFC 4.2

encoding and the Common Schema) at the top level, with sub-packages established to suit the needs of each project domain.

Figure 3 summarises the essential deliverables from this process. All the outputs of the work are brought together, first into a harmonised UML Report and then into a harmonised IFC schema that incorporates the infrastructure extensions.

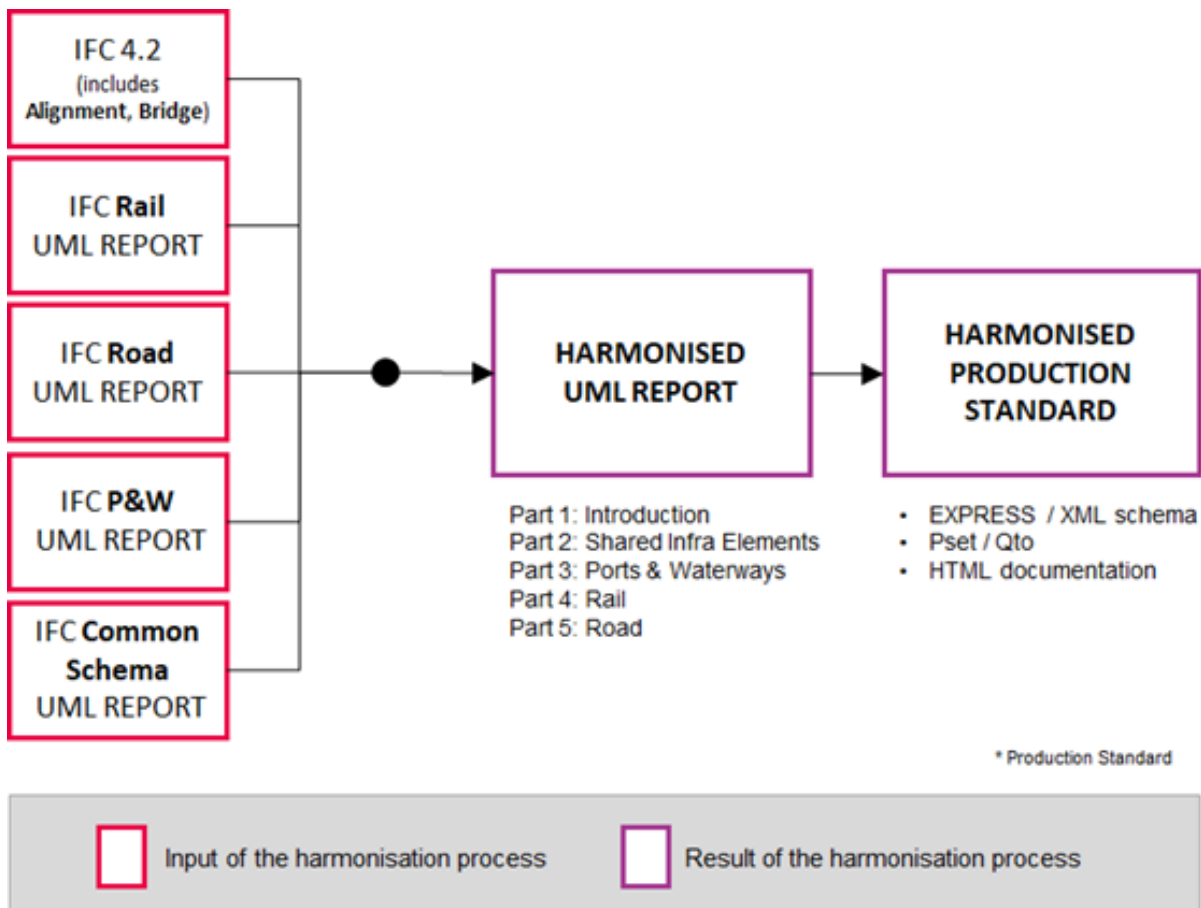


Figure 3 Outcomes of the Harmonisation Process

The deliverables shown in Figure 3 have been produced in a semi-automatic fashion from the harmonised schema held in the shared cloud environment: the four technical reports (Parts 2-5) are based on a common template; the fully-harmonised schema has been exported in XMI format and pushed into the ifcDoc repository.

A critical part of this process was the development of UML Modelling Guidelines (see Appendix A) that describe the way the UML model has been built, with the general rules and patterns that have been followed. This ensures consistency, facilitates the publication process and clarity of the UML diagrams presented in this report, and smooths the transfer process from the UML model (exported in XMI format) to the ifcDoc repository. It is important to note that while 95% of the guidelines are appropriate to UML as a general-purpose conceptual modelling language, there are parts that make use of constructs supported

specifically by Enterprise Architect, employed here to facilitate the harmonisation process and conversion to an IFC specification held in the ifcDoc repository.

The ifcDoc repository allows the schema (with infrastructure extensions) to be specified in EXPRESS and XML format and the normal HTML documentation published to the Web (including the change log that identifies all the extensions and proposed changes to the IFC schema). There will still need to be some refinements to the documentation to include information not held in the UML models, but the entity descriptions and relationships will be defined automatically. As a final step in the process, the ifcDoc web documentation will be checked by the domain teams to ensure the integrity of the harmonisation process and add that extra information (formal propositions, diagrams, etc.). The Property and Quantity Take-Off sets are held separately and will become part of the final documentation on ifcDoc as they are delivered by the separate domain projects.

Table 1 summarises the size of the proposed infrastructure extensions in terms of new entities, predefined types (PDTs), Enumeration Types, SELECT types, new type entities, modified entities and deprecated concepts. This is broken down into project domains but should not be interpreted as a measure of the impact of those domain projects on the IFC schema since many domain concepts were shifted to the Shared Elements during the harmonisation process.

Table 1 Extension Schema Statistics

	Total	Shared	P&W	Rail	Road
New Occurrence Entities	44	20	5	13	6
New Predefined Types	286	68	58	121	39
New Type Entities	14	5	4	5	0
New Select Types	5	5	0	0	0
New Enumeration Types	25				
Deprecated Entities	24				
Modified Entities	34				

3 Next Steps

The harmonisation process has focussed on the proposed IFC entities (new and modified existing) and associated predefined types. Some domain Property Sets have been specified as part of the domain projects, but much of that remains as a work in progress. An important next step in harmonisation (deliberately postponed until 2020) is to identify and resolve conflicts and duplication of terms in those property sets.

The ifcDoc repository will be important as we move to the next stage in this work, taking the harmonised schema and validating it through a rigorous deployment and software implementation process based on identified test cases. That work is planned for 2020 and is the subject of a separate project proposal.

4 Scope of the Harmonised Schema

The following sections provide a general indication based on initial scope statement of each domain project, showing also what is out-of-scope at this stage.

4.1 Rail Project

Domain Track

In-scope

- Panels (Track, Turnout, Dilatation)
- Objects of Track (Rail, Sleeper, Fastening)
- Ballast
- Slab track
- Rack Railway
- Elements of turnouts
- Track covering (for level crossing, light rails, tramways)
- Track alignment stops like buffers
- Track bench
- Lubrication
- Special equipment for shunting yards
- Track spatial structures
- Survey element
- Alignment

Out of scope

- Subsoil (should be treated by ifcRoad)
- Underground (should be treated by ifc Earthworks)
- Drainage of track
- Temporary objects
- Functional views and conditions
- Special equipment for depots (turntable etc.)

Domain Energy

In-scope

- Substations
- Earthing and current return
- Overhead constructions and supporting structures
- Overhead lines
- Switching post
- Suffix post
- Foundation and Fundaments
- AC and DC Installations
- Protection devices (Birds, touch protection)
- Lineside signs and signals

Out of scope

- High voltage lines (Distribution network)
- Power plants
- Rigid catenary
- Catenary for Tramways and light rails
- Trolley bus overhead lines
- Induction lines (non-contact system)
- Third rail (mounted trackside / on track panel)
- Equipment for diesel powered trains
- Equipment for steam powered trains
- Equipment for gas powered trains
- Installations for consumption measurements

Domain Signalling

In-scope

- Lineside installations
- Main signals (as standalone objects, simplified modelled)
- Shunting signals
- Relays
- All types of trackside signals and signs as information for train driver (no specific function yet)
- ETCS/CTCS lineside equipment (Balises, signs etc)
- Barriers for level crossing
- Warning signals at level crossings for road and pedestrian traffic (lights and bell)
- Level cross protection signs for rail traffic
- Operation and Surveillance equipment (Computer, Cabinets, Video cameras)
- Turnout machines and mounting installation, incl. manual switch lever
- Turnout heating (only electrical)
- Signalling cables (incl. trench, cable canal)
- Trackside sensors (Hotbox, etc.)
- Axle counter

Out of scope

- Signal components (like aspect lamps etc.)
- Logical and functional detailed aspects
- Small electronical components (fuses, etc.)
- ETCS/CTCS on board equipment
- Mechanical signalling equipment (bares, steel cables)
- Rods / turnout lock
- Gas turnout heating
- Signals for tramways and light rails
- Natural hazards sensors/surveillance
- Dynamic axle weight device

Domain Telecom

In-scope

- Mainly trackside equipment
- Terminals
- Cabinets and shelters
- Cabling (cables and connectors)
- Cable Routing (Laying installations)
- Sensors (snow, wind etc.)
- Antennas
- Towers and Poles
- Active Networks
- Base Transceiver Stations (BTS)
- E-Utran Node B for LTE (4G)
- Lineside telephones
- Vending and ticket machines
- Tetra Networks (limited mobile network)

Out of scope

- Centrals (inside equipment)
- Servers, terminals, computers, consoles (inside equipment)
- Radio inside devices
- Operation and surveillance installations
- CCTV
- Security systems (Access controls, batch readers etc.)
- Customer information systems (screens, loudspeakers)
- Functional modelling

4.2 IFC Road

In-scope:

- Linear road types:
 - Controlled access highway;
 - Dual carriageway;
 - Single carriageway;
 - Street;
 - Bicycle path;
 - Footpath.
- Junction types:
 - Interchange (grade separated):
 - overpass;
 - underpass;
 - ramp.
 - Intersection (at grade):
 - Intersecting roads (3, 4, ..., 7 way);
 - roundabout or traffic circle;
 - pedestrian crossing;
 - bicycle crossing.
- Road components, elements and equipment: Some of these concepts may be identified as being common and handed over to the common schema project and some may be developed by the IFC Road project team for the Common Schema project.
 - Road structure (road prism (road body))
 - Road guard elements
 - Road sign elements
 - Road paving components
 - Utilities
 - Lighting, telecommunications and power
 - Storm-, surface- water and drainage systems
 - Other underground facilities located in the road body.

Expected to be covered (but not subject to validation tests):

- paved surfaces of:
 - parking lots;
 - service areas;
 - toll plazas;
 - parking buildings;
 - ferry ports;
 - airports.

Out of scope:

- Equipment and buildings of the above listed paved surfaces;
- railway crossings;
- tramways;
- city scape / urban planning.

The following developments are out of scope for IFC Road because they are delivered through the Common Schema project:

- Earthworks cut and fill design;
- Geotechnical investigations;
- Geotechnical constructions.

4.3 Ports & Waterways Project

In-scope:

- **Complex types**
 - Cargo ports (Container, dry bulk, liquid bulk, roll-on roll-off)
 - Passenger terminals (cruise, ferry & foot traffic)
 - Marinas & leisure terminals
 - Ship & boat building yards
 - Marine & water maintenance
 - Offshore support/construction base
 - Canal complexes/networks
 - Channel regulation & water control
 - Ship lock complexes
- **Facility types**
 - Breakwaters
 - Revetments
 - Sluices, spillways & control gates
 - Ship-lifts
 - Hydraulic lift docks
 - Slipways
 - Dry docks
 - Floating docks
 - Wharf/Quays & Piers/Jetties
 - Ship locks
 - Anchorages
 - Navigational channels/areas
 - Storage/working areas
 - Intermodal yards/areas
- **Marine Products & Components**
 - Cargo & Vehicles
 - Cranes
 - Marine dock/lock gates
 - Aids to navigation (buoys, lights, markers etc.)
 - Fenders & bollards
 - Mooring systems & devices
 - Rock armour systems

Expected but not validated

- **Complex types**
 - Lifeboat & coast guard stations¹
 - Military complexes¹
- **Facility Types**
 - Retaining walls
 - Port/Complex Roads²
 - Railways (for cranes & vessel transfer systems)²
 - Tunnels
 - Buildings³
- **Geology, hydrology & geotechnics⁴**
 - Geo strengthening
 - Hydrology/water elements
 - Earthworks

Out of Scope

- **Complex types**
 - Coastal protection
 - Erosion protection
 - Flood protection
 - Power generation (Hydroelectric, tidal, wave, offshore wind)
- **Facility types**
 - Seawalls
 - Groynes
 - Dams/levees
 - Weirs

1. Is considered in scope due to similarities with other complex types being addressed but will not be explicitly validated.
2. Roads, railways, bridges & tunnels are considered vital facilities within ports & waterways but are included as referenced facilities where their content will be developed by other existing projects.
3. Entrance gates and buildings form the use of existing functionality for buildings within the IFC, with an extension of the possible typing mechanism to provide identification and classification of maritime specific buildings.
4. Geology, hydrology & geotechnics features are key to the maritime domain but have been handed off to the common schema project for common development.

Appendix A – bSI UML Modelling Guidelines (DRAFT)

bSI UML Modelling Guidelines

General principles for the authoring & coordination of UML modelling across IFC domains & projects

Project/Publisher: Infra Extension Office (Common Schema)

Work Package: WP2 – Harmonisation

Author(s):

EA - Evandro Alfieri

Engisis

FH - Florian Hulin

SNCF Réseau

CM - Claude Marschal

R+P AG

LW - Lars Wikström

Triona

MP - Matthiew Perin

Railenium

AB - Alex Bradley

Cardiff University

CZ - Chi Zhang

Applitec

Date: 04/02/2020

Version: **V05** – DRAFT

Document Information

Document ID	Title	Created By	Created
IR-CS-WP2	UML Modelling Guidelines	LW	2019-10-07

Revision History

Version	Status	Lead	Date	Notes
V01	DRAFT	LW	2019-10-07	First draft derived from Rail UML Guidelines
V02	DRAFT	AB/LW	2019-11-18	PIM Guidelines
V03	DRAFT	AB/LW	2019-11-20	PSM Guidelines Draft
V04	DRAFT	AB	2020-01-28	Updates from comments and further modelling + Design Patterns
V05	Draft	AB	2020-02-04	Finals changes to Full draft for standards submission

DRAFT

CONTENTS

Introduction.....	21
1 Terms and Definitions	22
2 Overview of bSI UML Model.....	25
2.1 Foreword	25
2.2 General Overview	26
2.2.1 Use Case View	26
2.2.2 Structural View	26
2.3 Package Structures	27
2.3.1 Top Level Structure	27
2.3.2 Platform Independent Model (PIM) Structure.....	28
2.3.3 Platform Specific Model (PSM) for IFC Structure	31
2.4 Status Management	32
2.4.1 Status Types, Workflow & Colour Coding	32
2.5 General etiquette	34
2.5.1 Modelling etiquette.....	34
2.5.2 Diagram etiquette	34
3 Platform Independent Model (PIM).....	36
3.1 Foreword	36
3.2 Naming Conventions	36
3.3 Colour Coding	36
3.4 Elements.....	37
3.4.1 Objects.....	37
3.4.2 Properties	37
3.4.3 Enumerations	38
3.4.4 Datatypes.....	38
3.4.5 Descriptions.....	38
3.4.6 Tag Values.....	38
3.5 Relationships	39
3.6 General Modelling Tips	39
4 Platform Specific Model (PSM).....	40

4.1	Foreword	40
4.2	PSM Modelling Guidelines	40
4.2.1	Naming Conventions	40
4.2.2	EXPRESS & IFC Constructs	41
4.2.3	Colour Coding	46
4.2.4	Tag Values.....	47
4.2.5	Summary Table.....	48
4.3	Mapping Guidelines.....	48
5	Design Patterns	49
5.1	Status Design Patterns	50
5.1.1	SDP1 – New IFC Concept Proposed.....	51
5.1.2	SDP2 – Deprecation of existing IFC Concept	51
5.1.3	SDP3 - Modification to existing IFC concept and/or IFC Relationship.....	52
5.2	Extension Design Patterns.....	54
5.2.1	EDP1 - New IFC entity and/or entity type as a subtype	54
5.2.2	EDP2 - Modification to an existing IFC entity.....	55
5.2.3	EDP3 - New IFC attribute on existing/proposed IFC entity	56
5.2.4	EDP4 - Modified IFC attribute on existing IFC entity.....	57
5.2.5	EDP5 - New predefined type container connected to existing/proposed IFC entity	58
5.2.6	EDP6 – New predefined type on existing/proposed predefined type container.....	59
5.2.7	EDP7 - New IFC property set and/or IFC quantity set	60
5.2.8	EDP8 - Modelling of IFC select.....	61
5.3	Mapping Design Patterns	63
6	Annexes	64

INTRODUCTION

The current need to provide a uniform and efficient method for the extension of the BuildingSMART standards requires the definition of a set of guidelines and best practices for the modelling of IFC in a UML based environment. UML has been chosen as the most suitable medium for both the authoring and communication of the current and future domain standard extensions. In relation to these requirements the present document has two goals:

- 1.1. To enable the uniformity of the various UML models, via common guidelines, recommendations and practices to their editors.
- 1.2. To provide a guideline to domain experts to assist in reading and understanding the IFC UML Model produced by current and future domain extension projects.

This uniform approach is beneficial for many reasons. These include:

- **A one-for-all solution:** different recurring problems can be parsed with unique, consistent, robust solutions
- **Common semantics:** authoring, reading and interpreting models made by current and future domain projects/extensions is done with ease, via a common language and representation.
- **Tool chain:** a consistent and open model profile can be easily integrated into new and existing tools, for further efficient workflows and further processing.

- IN.1. This guideline sets out the rules, clauses and recommendations for MODELLING within the IFC UML model and approaches this in a software agnostic manner. Information regarding the implementation of the bSI common development environment is set out in Annex A.1 of this document.
- IN.2. All rules, clauses and recommendations within this document apply to all bSI standard extension projects. Further rules and guidelines can be elaborated for each project and/or domain via domain annexes that accompany this document.

***NOTE** some figures contained in the following guidelines may refer to the specific tool used in domain projects (Enterprise Architect [EA], by Sparx Systems). This does not affect by any means the principles and the semantics of UML or restrict the use of other UML tools.*

1 TERMS AND DEFINITIONS

1.1 BIMQ

An information management tool adopted by the bSI Infrastructure Room and Railway Room

1.2 bSI UML model

This refers to the entire UML model. It includes all current project extensions which can include their platform independent and platform specific models (PIM & PSM)

1.3 Business object

A business object refers to any generic object be it physical, spatial or virtual that is relevant to the built environment or domain model it is defined in (e.g. rail switch, bridge pillar, road signal, marine fender, etc). Not intended to relate to the *IfcObject* templates, unless otherwise stated.

1.4 Concept

A concept is an abstract idea and varies by the context it is used in:

- **In IFC context:** it indicates any IFC concept (e.g. *IfcDoor*, *IfcWallType*, etc)
- **In UML context:** it indicates any UML Element (i.e. Class, Interface, Enumeration, etc.)

1.5 Conceptual model

In the scope of this document, conceptual model refers to the business model of domains, independently from IFC. This may also be referred to as a PIM (see Platform independent model).

1.6 Enterprise Architect (EA)

A unified modelling language (UML) environment and tool currently adopted by the bSI Infrastructure Room and Railway Room for the development of industry foundation classes (IFC) domain and common extensions

1.7 EXPRESS

EXPRESS is a standard data modelling language for product data and is formalised by the ISO10303 series of standards (specifically ISO10303-11).

1.8 Extensible mark-up language (XML)

Extensible Mark-up Language (XML) is a standard mark-up language. It is a W3C Recommendation.

1.9 Industry Foundation Classes (IFC)

IFC is a standardized, digital description of the built asset industry. It is an open, international standard (ISO 16739-1:2018) and promotes vendor-neutral, or agnostic, and usable capabilities across a wide range of hardware devices, software platforms, and interfaces for many different use cases.

1.10 IFC data type

IFC data types are the low-level definitions of value data within the IFC. These represent constrained numbers, labels, descriptions and matrices. In addition, they represent standardised measured values such length measures, ratios, speeds and mass all which have associated units. These objects are generally defined as an EXPRESS DEFINED TYPE.

1.11 IFC entity

IFC entities represent all primary (modelled) entities that derive from *IfcRoot* and all resource entities these objects are defined as an EXPRESS ENTITY.

1.12 IFC enumeration

IFC enumerations are finite lists of strings that provide a defined set of values to a context. These concepts are defined as an EXPRESS TYPE ENUMERATION.

1.13 IFC predefined type (PDT)

IFC Predefined types exist as more specific classifiers of generic types. These are represented in EXPRESS as enumeration lists or enumeration hierarchies and linked to the generic type via a property (usually called *PredefinedType*). See section 4.2.2.3 for further details

1.14 IFC <Project> model

This refers to the combination of the Conceptual Model plus the IFC extension to the business model for a project. It includes existing and proposed IFC entities.

1.15 IFC select

An IFC Select is simply a substitution group within the IFC. These are defined as an EXPRESS SELECT

1.16 IFC select hierarchy

An IFC select hierarchy is the combination of IFC selects and predefined type containers which represents a hierarchy of possible predefined types. See section 4.2.2.6 for further details.

1.17 IFC specification

A combination of 3 elements that defines and documents a version of the IFC standard. These are:

- EXPRESS Schema (captured & derived from the PSM of the bSI UML Model)
- Property/Quantity set definitions (captured & derived from the PSM of the bSI UML Model)
- IFC Documentation - entity definitions, formal propositions, notes, diagrams captured as HTML and PDF documentation (partially derived from the PSM)

1.18 Platform independent model (PIM)

A business model of domain(s) independent from IFC and any other schema specification.

1.19 Platform specific model (PSM)

A model that makes use of IFC for its realization and extension to cover the needs and requirements of the associated platform independent model (PIM)

1.20 Property set (Pset)

Property sets (Pset) are custom properties for concepts in IFC that may be applicable internationally, regionally, or specific to organization or project. A basic set of international Pset's are documented in the IFC specification as reference data.

1.21 Quantity set (Qto)

Quantity sets (Qto) are specific types of properties, which are quantity features (e.g. length, area, volume) of concepts. A basic set of international Qto's are documented in the IFC specification as reference data.

1.22 Unified modelling language (UML)

Unified Modelling Language (UML) is a standard language defined to specify, visualize and document models in software systems. It is defined by Object Management Group (OMG) and by ISO/IEC 19505:2005.

DRAFT

2 OVERVIEW OF BSI UML MODEL

2.1 FOREWORD

The BuildingSMART UML Model represents a part of the evolution of techniques for the development, representation and conveyance of BuildingSMART standards in a computer interpretable, modern and repeatable development environment. The bSI UML model provides the ability to utilise; standardised UML development environments to progress the model and widely adopted and easy to understand graphical notation to communicate and review changes.

The bSI UML model allows the capture of multiple views (elaborated in next sections) such as use cases and the structural data model within its structure. These views can be interlinked to convey further information and semantics. The structural view has a unique feature related to the definition of a dual soul of model development. The structural view addresses 2 types of sub-mode, a platform independent model (PIM) and a platform specific model (PSM-IFC). The PIM is developed to represent a true conceptual model of a given scope and business domain and is not tied to any modelling language or style. The PIM is an unconstrained model allowing the capture of domain knowledge from experts with no prior knowledge of particular modelling constructs (in the case of bSI this refers to the Industry Foundation Classes (IFC) and other bSI standards). This PIM can be encoded in UML within this model or can utilise other technologies (e.g. semantic web technologies). Whereas the PSM-IFC is tightly governed by the concepts, constructs and modelling language of the Industry Foundation Classes, representing the specific structure of the IFC schema in UML. This allows the graphical definition of the IFC extensions and most importantly the mapping of PIM concepts to specific IFC concepts for the purpose of encoding the business model.

It is hoped that this methodology is extended to further define and encode all existing concepts, constructs and modelling languages currently utilised within BuildingSMART standards to allow the integrated development of standards for the built environment.

2.2 GENERAL OVERVIEW

A UML model is typically divided into views that capture the different perspectives that need to be modelled. The expected views that will be utilised within the IFC UML model are the use case view & structural view.

2.2.1 Use Case View

The use case view encompasses the models which define a solution to a problem as understood by the client or stakeholders. This view encompasses the following elements:

- **Use Case diagrams** that depict the functionality required by the solution and the interaction of users and other elements (known as actors) with respect to the specific solution.
- **Use case documentation** that elaborates the key information around each use case such as atomic processes, geometry representations & semantic representation.

Inclusion of a use case model within the IFC UML model for each domain extension is **OPTIONAL**. Further standardization work in BuildingSMART is required to guide the definition of use cases within the UML model.

2.2.2 Structural View

The structural view encompasses the models which provide the static, structural dimensions and properties of the modelled solution. Elements applicable to this view include:

- **Class Diagrams** describe the static structure and contents of a solution using elements such as classes, interfaces and packages to display relationships such as containment, inheritance and associations. These diagrams are **REQUIRED**
- **Object Diagrams** depict a class or the static structure of a solution at a point in time. This can also be referred to as an instance diagram. These diagrams are **OPTIONAL**
- **Package Diagrams** depict the higher-level relationships and structure between packages within the UML model. These diagrams are **OPTIONAL**

In the context of the IFC UML model each domain under development is composed of 2 primary packages modelled using the structural view. These are:

1. **Platform independent model (PIM):** a conceptual model that represents the business model of that domain, independent from IFC. This element is **OPTIONAL**.
2. **Platform specific model (PSM):** represents the IFC extension of the business model. This element is **REQUIRED**. It is realized by two activities:
 - 2.1. **IFC extension proposal:** in which extensions to the existing IFC schema are proposed, based on business need. This element is **REQUIRED** *when a PIM is included*
 - 2.2. **IFC mapping,** in which the business objects and relations defined in the PIM are mapped toward the IFC schema, both the existing and the proposed extension. This element is **REQUIRED**.

NOTE: The **PIM** and the **PSM** are always referred to as two different entities, although they are contained in one domain package or file.

2.3 PACKAGE STRUCTURES

The following section provides an overview of the package structure within the entire IFC UML model.

NOTE: Package names should be as short and succinct as possible to reduce string lengths on fully qualified objects names. This is applied to both the PIM and PSM-IFC.

2.3.1 Top Level Structure

The IFC UML model's structure is depicted in Figure 1, and defined according to the listing below:

- **Model:** The root of the entire IFC UML model
 - 👁 **Class view:** A view package for the structural view to separate from other potential views. This is only required when more than one view is utilised within the model.
 - **IFC specification** includes all implemented IFC concepts in current baseline standard.
 - **Package overview:** a package diagram showing the top-level packages and their dependencies. This is **OPTIONAL**.
 - **Projects:** Sub package grouping together all active projects.
 - **<Project>**: The project specific concepts per domain (e.g. Common Schema, Rail, Road, Ports and Waterways) where Common Schema contains concepts which are agreed and harmonized on an overarching level. Within each domain package there are 2 main packages:
 - **PIM:** which contains the Platform Independent Model for the domain. This package may be additionally divided into sub packages as required by the domain.
 - **PSM-IFC:** which contains the Platform Specific Model for IFC. This package may be additionally divided into sub packages as required by the domain. This package includes definitions of all extensions and/or mappings to IFC as required by the domain.
 - 👁 **Use case view:** A view package for the use case view. Domain packages are **OPTIONAL**.
 - **Package overview:** a package diagram showing the top-level packages and their dependencies. This is **OPTIONAL**.
 - **<Project>**: The domain specific use case models with an additional Common package to illustrate generic and base use case definitions.

NOTE: Current baseline for IFC is 4.2. A possible reorganisation of the groupings within the standard could allow for an update to the top-level structure.

NOTE: If a domain does not have a platform independent model (PIM) the PIM & PSM package layer may be omitted and the sub packages that would be in the PSM will sit directly under the <Domain> package.

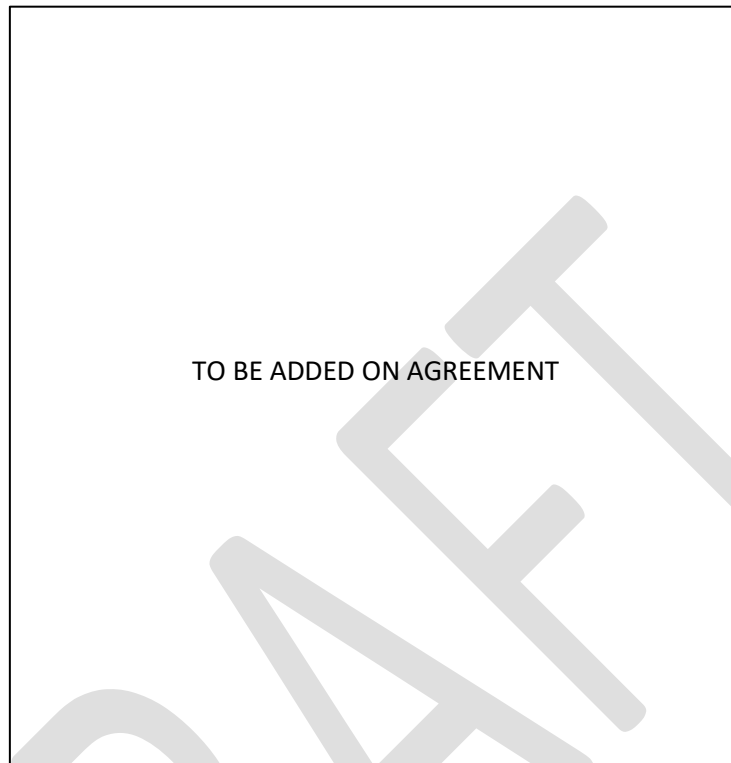


Figure 1 - Package structure of the IFC UML Model

Upon reaching and submitting a candidate standard for the next iteration of the platform specific model (IFC). All entities marked as ‘Candidate’ standard should be merged into the new ‘baseline standard’ within the relevant schema package. Once fully approved status is moved to implemented and version number incremented.

2.3.2 Platform Independent Model (PIM) Structure

The PIM per domain is free to be structured according to the requirements of that domain. However, the following is a **RECOMMENDATION**, which should be followed unless extenuating circumstances are present within the domain requirements.

If necessary, the PIM is sub-divided into packages according to clearly identified sub-domains based on how domain expertise is organized into disciplines. When sub-domain packages are defined an additional package named “Common” (or “Common – Shared”) must be included to contain the shared and common concepts/topics identified. An example can be seen in Figure 2.

Each business object is a unique entity inside the conceptual model, meaning it is described one time only. However, the same business object can be seen under multiple points of view. To facilitate the

reading and comprehension of the conceptual model, it is recommended to sub-divide each (sub-) domain into a spatial, structural (physical) and functional view:

1. **Spatial:** how objects occupy the space;
2. **Structural (physical):** how objects are (de)composed;
3. **Functional:** how objects are linked together and why.

Concepts are contained (owned) within one view (spatial, structural or functional) according to their definition type and then referenced when needed to identify relationships such as aggregation or connection. Table 1 provides an in-depth description of the objects and relationships defined in each view, and the expected references to object contained in other views.

The organization of the content of these sub-packages (spatial, structural and functional) is left to domains' internal use. However, it is **RECOMMENDED** these packages should contain the following:

- **Structural:**
 - All physical concepts.
 - structural diagrams to express physical breakdown of components, if needed.

NOTE: these concepts are candidates for IfcElement, IfcElementAssembly or IfcElementComponent
- **Spatial:**
 - only spatial concepts.
 - spatial diagrams to express the spatial breakdown of components, which can contain links to physical objects from the structural package usually contained within spatial concepts

NOTE: these concepts are candidates for IfcSpatialElements
- **Functional:**
 - only functional concepts.
 - functional diagrams to express the functional breakdown of components, which contains links to physical objects from the structural package.

NOTE: these concepts are candidates for IfcSpatialElements

Figure 2 depicts an example of how the platform independent model should be structured according to previous recommendation. In addition to the 3 views, a “Workbench” package is defined to contain all Work-In-Progress items and discussion within domain packages, to aid in status management and publishing of models.

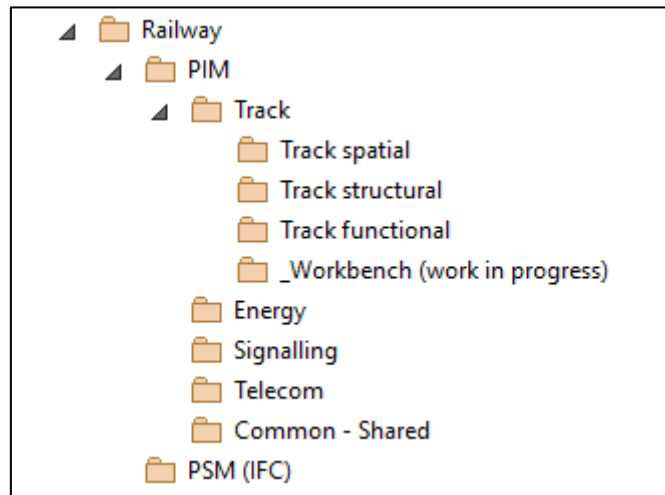


Figure 2 - Railway example for domain platform independent model (PIM) structure

Table 1 - Content descriptions for views within the PIM

View	Used to express	Note
Spatial	<p>Objects Places, volumes or zones where structural objects are located within the breakdown of the asset or project</p> <p>Relationships Aggregation of spatial objects Generalization of spatial objects</p>	The spatial layer may be sub divided if needed
Structural	<p>Objects Physically present and installed concepts that can be elements, assemblies or parts.</p> <p>Relationships Aggregation (i.e. part of) of physical objects Generalization of physical objects</p>	Structural objects are associated with properties to describe them further.
Functional	<p>Objects Functional spaces/areas where objects are gathered under a certain criterion. Functional groups where objects are collected under a certain criterion.</p> <p>Relationships Generalization of functional objects. Containment of structural/spatial objects. Connections (physical or virtual flow) of structural objects within Functional object.</p>	These views allow the organization in a functional and operation manner.

2.3.3 Platform Specific Model (PSM) for IFC Structure

The 'PSM for IFC' per domain is free to be structured according to the requirements of that domain. However, the following is a **RECOMMENDATION**, which should be followed unless extenuating circumstances are present within the domain requirements.

The PSM for IFC is sub-divided in packages according to the taxonomy in IFC, to aid in comprehension and relation to the base IFC specification. These packages are as follows:

- Physical Elements
 - Built Elements - *derives from IfcBuiltElement*
 - Distribution Elements - *derives from IfcDistributionElement*
 - Assemblies - *derives from IfcElementAssembly*
 - Components - *derives from IfcBuiltElement*
 - Feature Elements - *derives from IfcFeatureElement*
 - Furnishing Elements - *derives from IfcFurnishingElement*
 - Geographic Elements - *derives from IfcGeographicElement or IfcGeotechnicalElement*
 - Transport Elements - *derives from IfcTransportElement*
- Ports - *derives from IfcPort*
- Spatial Elements - *derives from IfcSpatialElement*
 - Spatial Structure Elements
 - Spatial Zones
- Systems - *derives from IfcGroup or IfcSystem*
 - Built Systems - *derives from IfcBuiltElement*
 - Distribution Systems - *derives from IfcDistributionElement*
- Positioning Elements
 - Alignment
 - Referent
- Geometric Representation and Placement

These packages should contain new IFC concepts, Property set definitions, Quantity set definitions and relationships that relate to the purpose of the division. An example is depicted in Figure 3. In addition, any modifications to the implemented IFC concepts should be noted by the change of status and their inclusion in the relevant diagram.

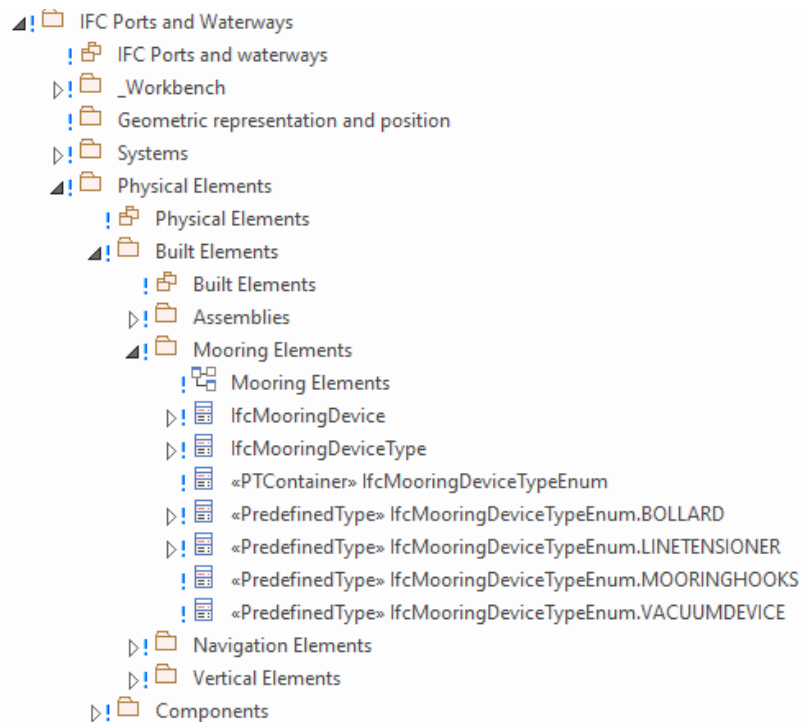


Figure 3 – example for platform specific model (PSM-IFC)

2.4 STATUS MANAGEMENT

To both manage the development and modification of concepts within the model and communicate these developments in diagrams a simple workflow and colour coordination is required. For design patterns illustrating the use of status see section 5.1.

2.4.1 Status Types, Workflow & Colour Coding

To manage status of Concepts (both in the PIM & PSM) the following status definitions shall be used:

- **WIP:** applies to elements that are in development and not fit for review.
- **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).
- **Approved:** applies to elements that have gone through expert panel review.
- **Candidate:** applies to elements that are included in a candidate standard release.
- **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.
- **Deprecated:** applies to element that have previously been implemented but are now marked for removal from the standard
- **Annotation:** applies to any other elements that do not relate to the structural modelling of the IFC UML model (e.g. notes & annotations).

NOTE: the 'WIP' status can be applied to any elements within the UML model, but is intended to be used on packages and classes that need to be identified as work in progress and excluded from generated documentation.

These status definitions are expected to change over the lifecycle of a concept as shown in Figure 4. To allow these status definitions to be visible on diagrams the legend & colour coding defined in Table 2 and depicted in Figure 4 will be used. Status will be identified by the colouring of the element border with a thickness of 2 points (or twice the default).

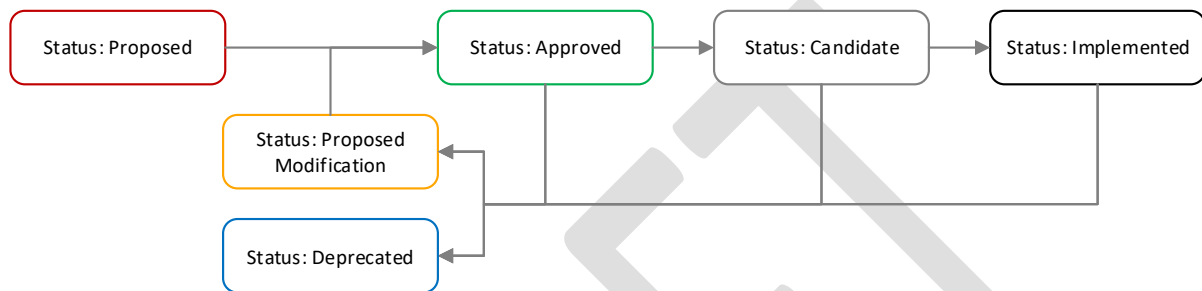


Figure 4 - Workflow for concept status evolution

The status of an element is used to flag new additions to the model and modifications to the baseline. In addition to the element status the following stereotypes can be used to further identify changes:

- **Modification to an attribute and/or relationship** – in the case where an attribute and/or relationship is modified such as a name change or type change, this shall be marked with a <<modified>> stereotype on the attribute and/or relationship along with a Proposed Modification status on the parent UML element. See section 5.1
- **Removal of an attribute and/or relationship** - in the case where an attribute and/or relationship is to be removed from a UML element, the attribute should not be deleted and instead marked with a <<Deprecate>> stereotype on the attribute and/or relationship along with a Proposed Modification (or Deprecation if the entire element is to be removed) status on the parent UML element. See section 5.1
- **Promotion of an attribute to supertype** – in the case where an attribute and/or relationship is to be promoted to a UML element higher in the inheritance tree. The moved attribute and/or relationship shall be marked with a <<promoted>> stereotype along with the appropriate statuses on the parent UML elements.
- **Demotion of an attribute to subtype** – in the case where an attribute and/or relationship is to be demoted to a UML element lower in the inheritance tree. The moved attribute and/or relationship shall be marked with a <<demoted>> stereotype along with the appropriate statuses on the parent UML elements.

Table 2 - Colour coding for element status definitions

Status Definition	Colour	RGB Definition (Decimal)
Proposed	RED	{192, 0, 0}
Proposed Modification	ORANGE	{255, 165, 0}
Approved	GREEN	{0, 176, 80}
Candidate	GREY	{127, 127, 127}
Implemented	BLACK (thickness: 1 or default)	{0, 0, 0}
Deprecated	BLUE	{0, 0, 205}

NOTE: WIP elements are not intended to be visually conveyed to a wider audience in forms such as diagrams and reports therefore does not require a colour coding or inclusion in legends.

2.5 GENERAL ETIQUETTE

2.5.1 Modelling etiquette

- Each bSI Room Project (i.e. IFC Rail) shall have a separate package in the IFC UML model as describe in section 2.3 above.
- If a Project (e.g. IFC Rail) is made of multiple domains, an integrated UML model shall be provided. In this case one package per domain shall be created (e.g. Track, Signalling, etc.).
- Models shall be clean and well organised. Detailed diagrams can help reading the model (see Diagrams etiquette below).
- Each UML model shall be accompanied by Baselines (XMI exports) of its content. This enables:
 - later model integration;
 - primitive but efficient model back-up;
 - basic compare/diff between different versions of the model;
- Frequency and strategy for Baselines generation shall be defined case by case;
- Packages shall be used to:
 - keep the model clean and well organised;
 - facilitate smart data migration between UML models and tools.

2.5.2 Diagram etiquette

- Diagrams shall be narrow in scope and well organised to enforce legibility. Large page format (i.e. ISO A0, A1, etc.) shall be avoided.
- Diagrams shall contain notes, especially if the model is shared among many subjects. Remember: "If it is not written, it does not exist".

- Diagrams shall always have a legend, especially the colourful ones.
- Diagram legends should be placed to one side of the diagram for efficient space usage.
- Diagram details (such as author, name & version) should be displayed on all diagrams.
- Elements present in the diagrams must follow the established colour-code. The rules are expressed in the next paragraphs.
- Each used element (including associations) shall be represented at least in one diagram.
- Workbench diagrams shall be used for temporary work and tests, not to be reviewed nor to understand the model.
- Workbench diagram shall be explicitly described as such.

DRAFT

3 PLATFORM INDEPENDENT MODEL (PIM)

3.1 FOREWORD

The purpose of the platform independent model (PIM) is to communicate and further define the data requirements of the domain. This model serves as:

- A representation of the concepts that define a business domain (i.e. Rail Track, Bridge, etc.).
- A reference for the proposal of IFC extensions.
- A reference for mapping a business domain toward IFC.
- A reference for any other systematic mappings (e.g. Requirements tools).
- Additional documentation to the IFC specification.

Also, it is important to note that:

- The platform independent model (PIM) IS NOT a replication of the data requirements expressed by domain groups, but a conceptualisation of them.
- A conceptual model IS NOT a Product Breakdown Structure (PBS).

The structure of the content shall follow the requirements and recommendations set out in section 2.3.2. this addresses the sub-division of the model and the semantics of the content. The purpose of this section is to set out rules for the representation of that content to aid in uniform representation and comprehension.

3.2 NAMING CONVENTIONS

- 3.2.1. All names, descriptions, notes of any element of the model shall be made in British English.
- 3.2.2. 'Ifc' and any other prefixes must be avoided in the conceptual model
- 3.2.3. Regular case shall be used (no *camelCase*, *PascalCase*, *snake_case*, etc.)
- 3.2.4. Duplicate names within the scope of the entire PIM for **different concepts** should be avoided unless necessary. This includes elements within different sub- packages.

3.3 COLOUR CODING

Colour coding is vital to the uniformity and comprehension of all models therefore the following clauses must be strictly adhered to. Semantic colour coding of elements is applied to the **BACKGROUND** of the object. Table 3 provides a summary of the following clauses.

*NOTE: It is **RECOMMENDED** that any auto colouring functions of the chosen development environment be utilized to aid in the efficient and uniform colouring of UML diagrams*

- 3.3.1. Spaces/zones/volumes (All spatial objects contained in the spatial view) are coloured with **WHITE** background

- 3.3.2. Functional spaces (volumes & spaces within the functional view) are coloured with **BROWN** background
- 3.3.3. Functional groups (systems & groups within the functional view) are coloured with **GREY** background
- 3.3.4. Alignment elements are coloured with **BLUE** background
- 3.3.5. Physical elements (physical elements within the structural view) are coloured with **ANY** background colour **EXCEPT** white, brown, grey & blue. Definitive colour coding and further division of physical elements can be defined in a supplementary domain annex.
- 3.3.6. Text colour of elements should be **BLACK**, except when background colouring hinders readability then colour of **WHITE** is applied

Table 3 - Colour coding for elements of the platform independent model (PIM)

Status Definition	Border Colour	RGB Definition (Decimal)
Spaces, volumes & zones	WHITE	{255, 255, 255}
Functional spaces	BROWN	{150, 75, 0}
Functional groups	GREY	{127, 127, 127}
Alignment elements	BLUE	{0, 0, 255}
Physical elements	ANY	--

3.4 ELEMENTS

Each element is a unique entity inside the platform independent model (PIM), meaning it is described one time only. However, to fully conceptualise and convey the purpose of the element the same object can be seen under multiple points of view.

3.4.1 Objects

- 3.4.1.1. Business objects are all modelled as UML Classes.
- 3.4.1.2. An element will only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 3.2.
- 3.4.1.3. **RECOMMENDATION:** An element's alias may be used to add terms in other languages.
- 3.4.1.4. Stereotypes are prohibited in the platform independent model (PIM)
- 3.4.1.5. All business objects required within a given business process or domain shall be present within the platform independent model (PIM). Only elements modelled within the PIM will be mapped and present within the Platform specific model (PSM-IFC).

3.4.2 Properties

- 3.4.2.1. Properties of business objects shall be included in the platform independent model (PIM)

- 3.4.2.2. Properties are represented as attributes on the relevant UML class
- 3.4.2.3. Property groups defined within domains are not reflected within the PIM.
- 3.4.2.4. A property's value type is applied using the TYPE field of the attribute. The value type is either an existing or proposed class within the model or Data type (see section 3.4.4).

3.4.3 Enumerations

- 3.4.3.1. Enumerations are modelled using a UML enumeration with each predefined value as an enumeration literal.
- 3.4.3.2. Sub-typing of business objects using enumerations is strictly prohibited in the platform independent models. A separate UML class should be defined with the appropriate generalisation relationship.
- 3.4.3.3. Enumerations shall only be used to represent a finite list of predefined values of a property.

NOTE: if the values of a property is unconstrained it is advised to use the appropriate data type.

3.4.4 Datatypes

- 3.4.4.1. Datatypes are modelled using the UML datatype element.
- 3.4.4.2. Data types are defined within the 'Data Types' package within the baseline IFC specification.
- 3.4.4.3. New data types are proposed within the relevant domain/project package and organised at the discretion of the domain. Addition/extension of data types is considered a last resort action, efforts should be made to utilise existing data types.
- 3.4.4.4. Data types are mapped towards IFC data types as these are already defined against established ISO standards.
- 3.4.4.5. Data types shall contain the following information
 - a. Name: A human readable identifier for the data type
 - b. Alias: A mapping towards the IFC implementation of the data type
 - c. Constraints: constraint rules associated with the use of the data type
 - d. Notes: A description of the data type for human definition
 - e. Link: Any links to related files or documents that define the data type.

3.4.5 Descriptions

- 3.4.5.1. Each element within the platform independent model shall have a description reported using the element note functionality
- 3.4.5.2. Descriptions reported shall be provide in British English.
- 3.4.5.3. Additional multilingual descriptions should be included through linked documents.

3.4.6 Tag Values

- 3.4.6.1. No restriction is placed on the use of tag values on elements.
- 3.4.6.2. **RECOMMENDATION:** Domains are urged to use tag values for the inclusion of additional identifiers for synchronization across different tools within the chosen tool chain.

3.5 RELATIONSHIPS

- 3.5.1. Relationships between business objects shall be defined using UML associations. **DO NOT** include any IFC implementation consideration in the PIM.
- 3.5.2. The allowed UML associations for the platform independent model (PIM) are described in Table 4.
- 3.5.3. Stereotypes are prohibited in the platform independent model (PIM)
- 3.5.4. Multiple inheritance is permitted in the platform independent model (PIM)
- 3.5.5. For each UML association there are two possible forms:
 - a. Generic (no text): to highlight that two concepts are somehow related
 - b. Described: to express constraints or additional context on the relationship, this is done via text, role names & multiplicity.
- 3.5.6. Multiplicity (cardinality) definition is **OPTIONAL** but encouraged as it aids in comprehension and mapping towards the PSM.
- 3.5.7. When multiplicity (cardinality) is omitted it is assumed to be unconstrained.

Table 4 - Relationships within the platform independent model (PIM)

Relationship	Definition & Recommendations
Aggregation	<ul style="list-style-type: none"> - Used to depict part-whole relationships between 2 or more objects. <p>RECOMMENDATIONS</p> <ul style="list-style-type: none"> - If linked to a property on source or target object use role names to associate the property to the association.
Generalisation	<ul style="list-style-type: none"> - Used to indicate inheritance. The source inherits the target's characteristics - Arrow points from <i>SPECIFIC</i> (source) classifier to <i>GENERAL</i> (target) classifier
Association	<ul style="list-style-type: none"> - A general relationship between elements - Used when none of the above are suitable relationships <p>RECOMMENDATIONS</p> <ul style="list-style-type: none"> - To provide further understanding, add a description and role names to associations. - If linked to a property on source or target object use role names to associate the property to the association. - If known, specify the multiplicity of the association roles.

3.6 GENERAL MODELLING TIPS

- 3.7.1. Each time there is disagreement on a definition, try to separate the object in two different concepts – just to help the discussion. Then, try to find a common ancestor (shared properties) to reduce the complexity of the model.
- 3.7.2. Do not try to have one object to express everything. Small is beautiful.

4 PLATFORM SPECIFIC MODEL (PSM)

4.1 FOREWORD

The purpose of the platform specific model (PSM) for IFC is to define and communicate the extension of the IFC. This model originates from two main activities, that run in parallel:

1. **The IFC extension proposal** where modifications and additions to the IFC specification are proposed;
2. **The IFC mapping**, in which business concepts/business objects are linked to existing/proposed IFC Concepts.

The structure of the content shall follow the requirements and recommendations set out in section 2.3.3. this addresses the sub-division of the model and the semantics of the content. The purpose of this section is to set out rules for the representation of that content to aid in uniform representation and comprehension.

If a platform independent model (PIM) is defined, the IFC mapping is **REQUIRED**, and should follow the rules and guidelines set out in section 4.3. All mapping definitions shall be included in any domain candidate standard submissions.

4.2 PSM MODELLING GUIDELINES

The following rules and guidelines describe how to model and represent new entities in the PSM-IFC. This guideline also provides descriptions for the reading of the IFC specification currently in implemented state which forms the basis when proposing modifications and additions

4.2.1 Naming Conventions

- 4.2.1.1. All names, descriptions, notes of any concept of the model shall be made in British English.
- 4.2.1.2. All names and identifiers shall use pascal case (e.g. IfcDoorType) unless otherwise specified
- 4.2.1.3. An **IFC entity** name shall be prefixed with the notation 'Ifc' followed by the unique identifying name. For example; **IfcDoor**.
- 4.2.1.4. An **IFC predefined type** (PDT) name shall be of the form <concept>TypeEnum.<PDT>. With the PDT value being all upper case, with no spaces. An example for the IfcDoor element is '**IfcDoorTypeEnum.GATE**'.
- 4.2.1.5. An **EXCEPTION** is applied in the case where readability is impaired on a multi word **IFC predefined type** (e.g. IfcMarinePartTypeEnum.**CILLLEVEL**) snake case is permitted to make the name easily understandable, an example is 'IfcMarinePartTypeEnum.**CILL_LEVEL**'.
- 4.2.1.6. An **IFC property set** name shall be prefixed with the notation 'Pset_' followed by the unique identifying name. For example; **Pset_DoorCommon**.

RECOMMENDATION: *The unique identifying name of a property set may follow one of these formats:*

- *Pset_<Entity>Common when applied directly to an IFC Entity, for example Pset_DoorCommon applied to IfcDoor.*
- *Pset_<Entity><PDT> when applied to a predefined type of an entity, for example Pset_SanitaryTerminalTypeSink applied to an IfcSanitaryTerminal with PDT SINK>*

- 4.2.1.7. An **IFC quantity set** name shall be prefixed with the notation 'Qto_' followed by the unique identifying name and suffixed with 'Quantities'. For example; **Qto_DoorBaseQuantities**. The unique identifying name should indicate the related entity and the type of quantities.
- 4.2.1.8. An **IFC enumeration** shall be suffixed with the notation 'Enum' to aid in identification. For example, IfcAssemblyPlaceEnum.
- 4.2.1.9. An **IFC Property set enumeration** shall be prefixed with the notation 'PEnum_' to aid in identification. For example, 'PEnum_ElementStatus'.
- 4.2.1.10. An **IFC select** is *recommended* to be suffixed with the notation 'Select' to aid in identification. For example, IfcInterferenceSelect.
- 4.2.1.11. An **IFC data type** shall be suffixed with the notation 'Measure' if such datatype will be instantiated as a numeric value with a measure unit, to aid in identification. For example, IfcPositiveLengthMeasure.

4.2.2 EXPRESS & IFC Constructs

This section provides guidelines and rules for the representation of IFC concepts in UML diagrams and models. Visual images can be found in the extension design patterns section by using the references in the summary table.

4.2.2.1 IFC Entities

IFC entities represent all primary (modelled) entities that derive from *IfcRoot* and all resource entities these objects are defined as an EXPRESS **ENTITY**.

- >.1. An IFC entity shall be modelled as a UML Class.
- >.2. An IFC entity shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC entity shall not have any stereotypes applied.
- >.4. An IFC entity shall define a description in the UML elements note to explain the semantics.
- >.5. IFC **strictly** enforces hierarchal inheritance therefore an entity can only inherit from one supertype. **Multiple inheritance is prohibited.**
- >.6. IFC entity inheritance is modelled using a UML generalisation where the arrow points from *SPECIFIC* (source) entity to *GENERAL* (target) entity.

4.2.2.2 Ifc Entity Attributes

An IFC entity can have an unlimited set of attributes assigned to it these provide information directly on the entity. These attributes can represent relationships between entities (reference type) or values categorised by data types (value type). These constructs are defined as **ATTRIBUTES** on an EXPRESS **ENTITY**.

- >.1. An IFC entity attribute shall be modelled as a UML Attribute on the UML Class
- >.2. An IFC entity attribute shall only have one primary name, and that name must be the English language name of the attribute and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC entity attribute shall have a type assigned to restrain the content of the attribute. This type **MUST** be an IFC entity, IFC select, IFC enumeration or IFC data type present within the Platform Specific Model (PSM).
- >.4. In the case when an IFC entity attribute's type is an IFC entity a UML association relationship shall be created with the appropriate source and target selected and multiplicity specified.
- >.5. An IFC entity attribute shall have a multiplicity defined this is made up of the following parameters; lower bound, upper bound, allow duplicates and ordered. If an accompanying UML association relationship is defined the multiplicity parameters shall match.
- >.6. An IFC entity attribute shall be marked with a scope of public. This is primarily for consistency across the model, as the concept of scope is not present in the EXPRESS modelling language.

4.2.2.3 IFC Predefined Types

IFC Predefined types exist as more specific classifiers of generic types. IFC adopts the practice of minimising entity count, therefore typing at a granular level is handled by Predefined types. These are represented in EXPRESS as enumeration lists or enumeration hierarchies and linked to the generic type via a property (usually called *PredefinedType*). In UML these values are modelled as full UML classes which is semantically correct and allows the association of property sets of IFC PDTs (the current functionality in IFC EXPRESS PSD-XML). Figure XX depicts the representation of these elements. The guidelines for the definition of a predefined type is as follows:

- >.1. An IFC predefined type shall be modelled as a UML Class.
- >.2. An IFC predefined type shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC predefined type may have an alias derived from the primary name's suffix after the period to aid in diagram layout and readability. For example, '*IfcDoorTypeEnum.GATE*' would have an alias of 'GATE'.
- >.4. An IFC predefined type shall have the stereotype <<PredefinedType>> applied.
- >.5. An IFC predefined type shall define a description in the UML elements note to explain the semantics.
- >.6. An IFC predefined type shall not have any attributes defined on the UML class, these should be encoded in IFC via property & quantity set, or if generic to all, via the parent IFC entity

- >.7. An IFC predefined type for *USERDEFINED* and *NOTDEFINED*, shall not be modelled it is **ASSUMED** that for each set of predefined types within a predefined type container that these PDTs are present. This is to improve readability and diagram clutter.

4.2.2.4 IFC Predefined Type Container

An IFC predefined type container is the IFC concept that groups together predefined types under a parent entity. It is associated to the parent entities via a UML association and related to the predefined types via a form of UML dependency. Predefined type containers are encoded in EXPRESS as enumerations or selects and should be named accordingly. Figure XX depicts the representation of these elements. The guidelines for the definition of a predefined type container is as follows:

- >.1. An IFC predefined type container shall be modelled as a UML Class.
- >.2. An IFC predefined type container shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC predefined type container shall have the stereotype <<PTContainer>> applied.
- >.4. An IFC predefined type container shall define a description in the UML elements note to explain the semantics.
- >.5. An IFC predefined type container shall not have any attributes defined on the UML interface.
- >.6. An IFC predefined type container that is a simple collection predefined types shall be related to those elements by a UML dependency relationship where the arrow points from *Container* (source) to *predefined type* (target).
- >.7. An IFC predefined type container that contains an enumeration hierarchy will use the construct defined in 4.2.2.6.
- >.8. An IFC predefined type container shall not contain an IFC predefined type for *USERDEFINED* and *NOTDEFINED*, these are **MANDATORY** within each container and therefore will be added during the conversion process to EXPRESS and other concrete modelling languages.

4.2.2.5 IFC Selects

An IFC Select is simply a substitution group within the IFC. Figure XX depicts the representation of these elements. The guidelines for the definition of a select is as follows:

- >.1. An IFC select shall be modelled as a UML Interface.
- >.2. An IFC select shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC select shall have the stereotype <<Select>> applied.
- >.4. An IFC select shall define a description in the UML elements note to explain the semantics.
- >.5. An IFC select shall not have any attributes defined on the UML interface.
- >.6. An IFC select shall be related to the substitution elements by a UML substitution relationship where the arrow points from *the Select* (source) to the *possible substitute*(target).
- >.7. An IFC select may substitute both data types and entities.

4.2.2.6 IFC Select Hierarchies

An IFC select hierarchy is the combination of IFC selects and predefined type containers which represents a hierarchy of possible predefined types. Therefore, the guidelines extend the rules laid out in 4.2.2.5 & 4.2.2.6. Figure XX depicts the representation of these elements. The guidelines for the definition of a select hierarchy is as follows:

NOTE: this is an experimental construct under review by technical experts. Its use is not recommended currently other than the test case of IfcFacilityPart predefined types.

- >.1. An IFC select in a hierarchy shall only substitute a predefined type container or another IFC select, this is applied recursively.
- >.2. The top select element is associated with the parent entity in the same way as a predefined type container, using a UML association.

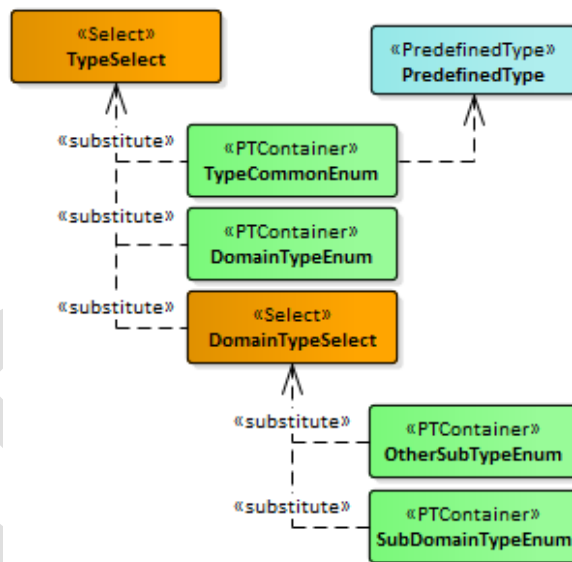


Figure 5 - Illustration of an IFC select hierarchy using selects and predefined type containers

4.2.2.7 IFC Enumerations

IFC enumerations are finite lists of strings that provide a defined set of values to a context. These concepts are defined as an EXPRESS TYPE **ENUMERATION**.

- >.1. An IFC enumeration shall be modelled as a UML enumeration.
- >.2. An IFC enumeration shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC enumeration shall have the stereotype <<enumeration>> applied.
- >.4. An IFC enumeration shall define a description in the UML element's note to explain the semantics.
- >.5. Enumerations shall only be used to represent a finite list of predefined values of a property.

4.2.2.8 IFC Data Types

IFC data types are the low-level definitions of value data within the IFC. These represent constrained numbers, labels, descriptions and matrices. In addition, they represent standardised measured values such length measures, ratios, speeds and mass all which have associated units. These objects are generally defined as an EXPRESS **DEFINED TYPE**.

- >.1. An IFC data type shall be modelled as a UML Datatype.
- >.2. An IFC data type shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC data type shall have the element type stereotypes <<Datatype>> applied.
- >.4. An IFC data type shall define a description in the UML elements note to explain the semantics.

4.2.2.9 IFC Property Sets

Property sets (Pset) are custom properties for concepts in IFC that may be applicable internationally, regionally, or specific to organization or project. A basic set of international Pset's are documented in the IFC specification as reference data. Property sets are defined in the specific Property set definition language (PSD-XML).

- >.1. An IFC property set shall be modelled as a UML Interface.
- >.2. An IFC property set shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC property set shall have the stereotype <<PropertySet>> applied.
- >.4. An IFC property set shall define a description in the UML elements note to explain the semantics.
- >.5. An IFC property set shall be related to the relevant IFC entity or predefined type by a UML realization relationship where the arrow points from *the property set* (source) to the *entity* (target).
- >.6. An IFC property set shall have a tag value called Pset Template with values of either *Type*, *Occurrence* or *Overridden*, to further define the property set
- >.7. An IFC property set's properties shall be modelled as attributes on the UML Interface.

4.2.2.10 IFC Quantity Sets.

Quantity sets (Qto) are specific types of properties, which are quantity features (e.g. length, area, volume) of concepts. A basic set of international Qto's are documented in the IFC specification as reference data.

- >.1. An IFC quantity set shall be modelled as a UML Interface.
- >.2. An IFC quantity set shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC quantity set shall have the stereotype <<QuantitySet>> applied.
- >.4. An IFC quantity set shall define a description in the UML elements note to explain the semantics.

- >.5. An IFC quantity set shall be related to the relevant IFC entity or predefined type by a UML realization relationship where the arrow points from *the quantity set* (source) to the *entity* (target).
- >.6. An IFC quantity set's properties shall be modelled as Attributes on the UML Interface.
- >.7. A quantity shall only have a value type of either an IFC data type or IFC resource entity. Relationships between IFC entities deriving from IfcRoot should use the current IfcRelationship concepts.

4.2.2.11 IFC Property Set Enumeration Types

An IFC property set enumeration type (PEnum Type) is an enumeration defined for use in relation to a property set. This is a uniquely IFC construct and is different to an IFC enumeration due to the division between the definition of IFC Concepts and IFC property sets. An IFC PEnum Type is defined in the specific property set definition language (PSD-XML), and is modelled in UML using the following clauses:

- >.1. An IFC PEnum Type shall be modelled as a UML Enumeration
- >.2. An IFC PEnum Type shall only have one primary name, and that name must be the English language name of the object and adhere to naming conventions in clause 4.2.1.
- >.3. An IFC PEnum Type shall have the stereotype <<PEnumType>> applied.
- >.4. An IFC PEnum Type shall define a description in the UML elements note to explain the semantics.
- >.5. An IFC PEnum Type shall be related to the relevant property set(s) via the typing of the property set attribute.
- >.6. It is **RECOMMENDED** that an IFC PEnum Type be related to the relevant property set(s) via a UML association relationship to further depict the connection.
- >.6. An IFC PEnum Type shall only be used to represent a finite list of predefined values of a property.

4.2.3 Colour Coding

Colour coding is vital to the uniformity and comprehension of all models therefore the following clauses must be strictly adhered to. Semantic colour coding of elements is applied to the **BACKGROUND** of the object. Table 3 provides a summary of the following clauses.

- 4.2.3.1. IFC entities (no stereotype) are coloured with **WHITE** background.
- 4.2.3.2. IFC predefined types (stereotype <<PredefinedType>>) are coloured with **LIGHT CYAN** background.
- 4.2.3.3. IFC predefined type containers (stereotype <<PTContainer>>) are coloured with **GREEN** background.
- 4.2.3.4. IFC virtual entities (stereotype <<VirtualEntity>>) are coloured with **VERY LIGHT CYAN** background.
- 4.2.3.5. IFC selects (stereotype <<Select>>) are coloured with **ORANGE** background.

- 4.2.3.6. IFC enumerations (stereotype <<Enumeration>>) are coloured with **LIGHT GREEN** background.
- 4.2.3.7. IFC property set enumerations (stereotype <<PEnumType>>) are coloured with **LIGHT ORANGE** background
- 4.2.3.8. IFC property sets (stereotype <<PropertySet>>) are coloured with **MAGENTA** background.
- 4.2.3.9. IFC quantity sets (stereotype <<QuantitySet>>) are coloured with **LIGHT MAGENTA** background.
- 4.2.3.10. IFC data types (stereotype <<Datatype>>) are coloured with **YELLOW** background.

Table 5 - Colour coding for elements of the platform specific model for IFC (PSM-IFC)

Status Definition	Colour	RGB Definition (Decimal)
IFC Entities	WHITE	{255, 255, 255}
IFC Predefined types	LIGHT CYAN	{179, 242, 242}
IFC Predefined Type Container	GREEN	{140, 255, 140}
IFC Virtual entities	VERY LIGHT CYAN	{217, 242, 242}
IFC Selects	ORANGE	{255, 165, 0}
IFC Enumerations	LIGHT GREEN	{204, 255, 204}
IFC property set enumeration	LIGHT ORANGE	{255, 218, 185}
IFC Property sets	MAGENTA	{221, 160, 221}
IFC Quantity sets	LIGHT MAGENTA	{255, 230, 255}
IFC Data types	YELLOW	{242,242,192}

4.2.4 Tag Values

- 4.2.4.1. No restriction is placed on the use of tag values on elements, unless explicitly mentioned within this document
- 4.2.4.2. An IFC property set shall have a tag value called Pset Template with values of either *Type*, *Occurrence* or *Overridden*, to further define the property set
- 4.2.4.3. **RECOMMENDATION:** Domains are urged to use tag values for the inclusion of additional identifiers for synchronization across different tools within the chosen tool chain. For example, an IfcDoc GUID.

4.2.5 Summary Table

Table 6 - Summary table for representation of IFC elements in UML form

Element	UML Element	Stereotype	Example Naming Convention	Notes	DP ¹
IFC Entity	Class	NONE	IfcNewEntity	Very 'expensive' for implementation. as few additions as possible	1, 2, 3
IFC Predefined Type	Class	<<PredefinedType>>	IfcNewEntityTypeEnum.NEW	Used for specific typing on defined entities. 'Less expensive 'to impl.	5, 6
IFC Select	Interface	<<Select>>	IfcSomeSelect	Useful for multiple inheritance and attributes relevant to multiple types	8
IFC Enumeration²	Enumeration	<<enumeration>>	IfcNewEnum	MUST follow enumeration rules and guidelines see X.XX	
IFC Property Set Enumeration	Enumeration	<<PEnumType>>	PEnum_NewThing	MUST follow property set enumeration rules and guidelines	
IFC Data Type	Data Type	<<Datatype>>	IfcNewMeasure	Additions should be well justified	
IFC Property Set	Interface	<<PropertySet>>	Pset_NewEntityCommon	Should be linked to relevant entities	7
IFC Quantity Set	Interface	<<QuantitySet>	Qto_NewEntityBaseQuantities	Should be linked to relevant entities	7

1. Design Patterns – references the relevant design pattern for an exemplar of representations
2. Not all IFC enumerations are presented as UML enumerations.

4.3 MAPPING GUIDELINES

IN DEVELOPMENT

5 DESIGN PATTERNS

Design patterns are general repeatable solutions to common occurring scenarios of the UML modelling activities. Design patterns are separated into 3 primary types:

- Patterns relating to use of status
- Patterns for the extension and modification of the PSM-IFC model
- Patterns for the mapping of PIM concepts to PSM concepts.

Design patterns are described according to the following structure:

- **Description:** A brief description of the objective and/or purpose of the design pattern
- **Steps:** how to implement the scenario in the general context of the bSI UML model.
- **Example diagram:** An annotated diagram pointing out the key requirements and elements of the design pattern.
- **Limitations of design pattern:** to clarify what are the limits of the developed solution

In addition to the above the extension design patterns section can include:

- **EXPRESS specification example:** an extract of EXPRESS to highlight how the design pattern is translated to the primary encoding format.
- **PSD-XML example:** an extract of PSD-XML to highlight how the design pattern is translated to the primary encoding format.

5.1 STATUS DESIGN PATTERNS

Status design patterns relate to the identification of changes to the bSI UML model, this is primarily applied to the Platform Specific Model (PSM-IFC) to track changes and enable efficient generation of encodings, but can also be applied to control the Platform Independent Model (PIM).

The status design patterns all use a singular example diagram focusing on different parts for each specific design pattern, the full diagram is shown in Figure 6. Within this section the following design patterns are elaborated.

- SDP1. New IFC concept proposed
- SDP2. Deprecation of existing IFC concept
- SDP3. Modification to existing IFC concept and/or IFC Relationship.

No design pattern is provided to cover statuses of; *Approved* and *Candidate* as these are meant to manage approval of the development work within individual project and the bSI standards process.

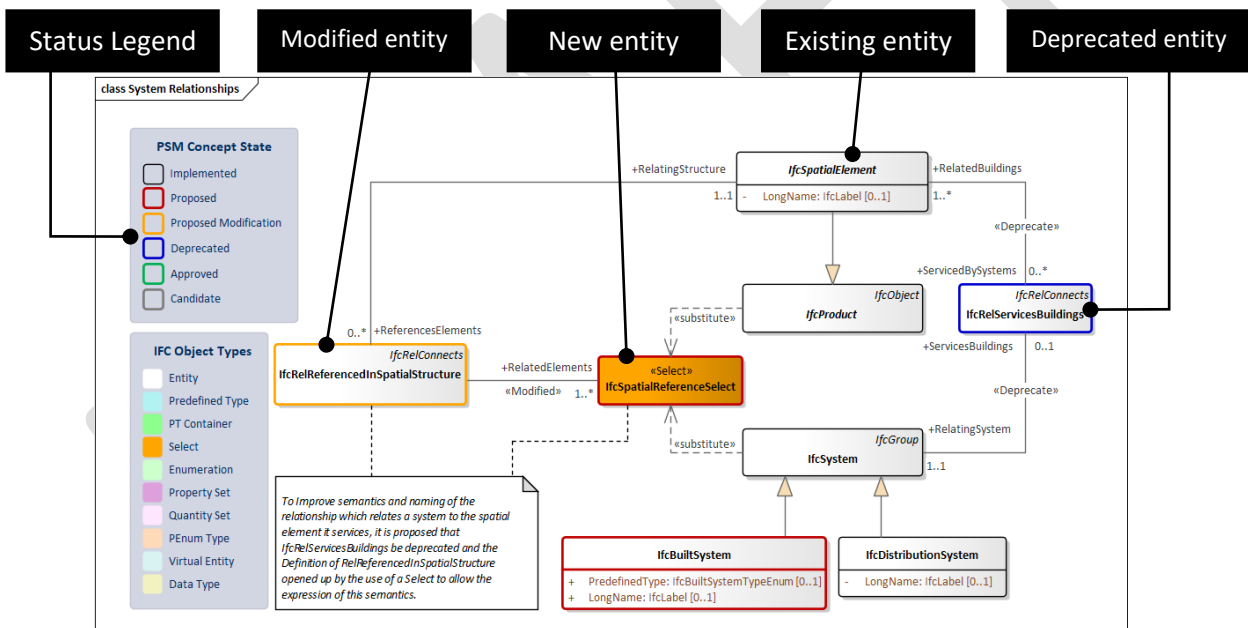


Figure 6 – Modifications and extensions to *IfcSystem* relationships - Status use example diagram

5.1.1 SDP1 – New IFC Concept Proposed

When a new IFC concept needs to be added to the bSI UML model its status must be marked correctly within the model to be conveyed to the wider public and international consultation involved in standards development. This example (Figure 7) shows the addition of a new IFC Select type *IfcSpatialReferenceSelect*, which creates a substitution relationship with existing entities *IfcProduct* and *IfcSystem*. Also, a new Ifc entity with its attributes, *IfcBuiltSystem* is defined with its inheritance relationship defined.

Typical Steps

1. Inside the relevant package, locate the diagram to illustrate the new IFC concept, or create the relevant diagram required.
2. Create and insert the new IFC concept and **set its status to *PROPOSED***, and set its name

RECOMMENDATION: *it is recommended to set the default status of UML elements to **PROPOSED** within the chosen development environment, minimizing the chance of orphaned and untracked additions.*

3. Author the relevant attributes and relationships required by the new IFC concept in line with the relevant modelling guidelines.

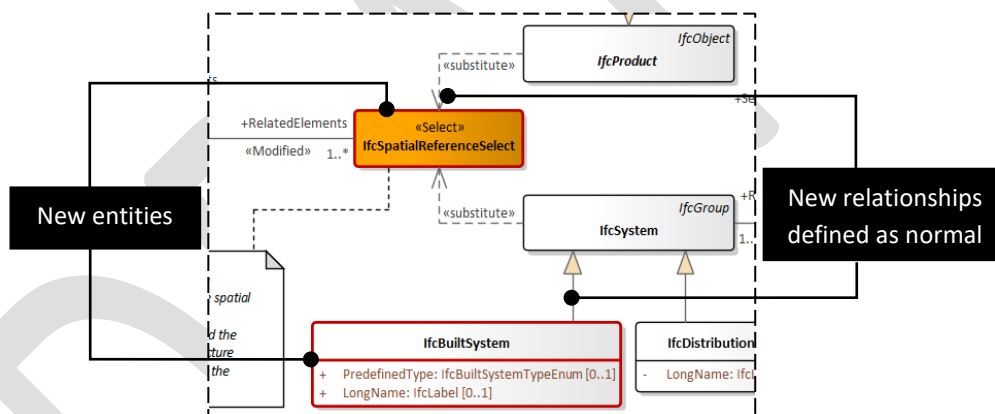


Figure 7 - Status use for new IFC entities



Limitations of SDP1

- Requires a dedicated status property (or defined tag pair) within development environment
- Works best with auto-colour functionality and legends.

5.1.2 SDP2 – Deprecation of existing IFC Concept

When an existing IFC concept (and associated elements) needs removed from the bSI UML model its status is changed to **DEPRECATED** within the model to convey the change to the wider public and international consultation involved in standards development. This example (Figure 8) shows the deprecation of an IFC Entity *IfcBuildingSystem* and associated attributes through a status change. Additionally, the existing relationships are marked using a `<<Deprecate>>`.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the deprecation of the IFC concept, or create the relevant diagram required.
2. Drag/Insert the existing IFC concept from the IFC specification baseline, additionally insert any existing IFC concepts that have relationships to the concept to be deprecated
3. Mark the IFC concept as **DEPRECATED** using status property
4. Mark any relationships with a <<Deprecate>> stereotype to highlight their removal.

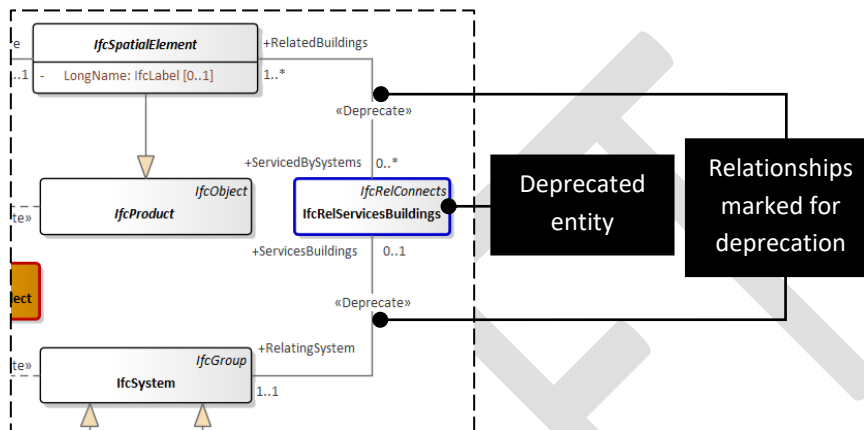


Figure 8 - Status use for deprecated entities and/or relationships

Limitations of SDP2

- Requires a dedicated status property (or defined tag pair) within development environment
- Works best with auto-colour functionality and legends.
- Relationships are flagged using a different mechanism (stereotyping) to IFC concepts

5.1.3 SDP3 - Modification to existing IFC concept and/or IFC Relationship

When an existing IFC concept (and associated elements) needs to be updated in the bSI UML model its status is changed to **PROPOSED MODIFICATION** to convey the change within the model to the wider public and international consultation involved in standards development. This example (Figure 9) shows the modification of the relationship (and source attribute), *relatedElements* on IFC entity *IfcRelReferencedInSpatialStructure*. The modified relationship is marked by the <<Modified>> stereotype. Additionally, the source concept of the relationship (and source attribute parent) is marked with the status change.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the modification of the IFC concept and/or relationship, or create the relevant diagram required.
2. Drag/Insert the existing IFC concept(s) from the IFC specification baseline, that will be modified or is the source/target of a relationship to be modified.

3. Make the required modifications to the model, attribute changes, relationship changes and/or description changes.
4. Mark the IFC concept (or source of modified relationship) as **PROPOSED MODIFICATION** using the status property
5. Mark any relationships with a <<Modified>> stereotype to highlight their change.

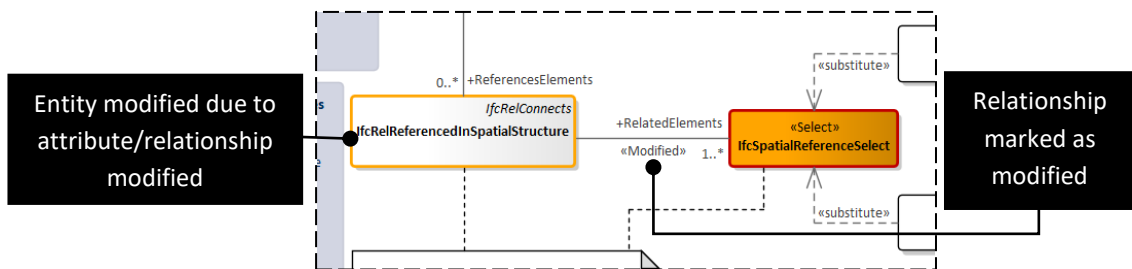


Figure 9 - Status use for modified entities and/or relationships

Limitations of SDP3

- Requires a dedicated status property (or defined tag pair) within development environment
- Works best with auto-colour functionality and legends.
- Relationships are flagged using a different mechanism (stereotyping) to IFC concepts
- Multiple properties need to be set to convey simple changes (e.g. change of attribute and relationship type). This is for readability and easier identification in automation tasks.

5.2 EXTENSION DESIGN PATTERNS

Extension design patterns relate to the provision of proposed and modified concepts within the platform specific model (PSM-IFC). The modelling of the PSM is critical due to the automation being provisioned onto the UML model to generate the different encoding formats. Within this section the following design patterns are elaborated.

- EDP1. New IFC entity and/or entity type as a subtype
- EDP2. Modification to an existing IFC entity
- EDP3. New IFC attribute on existing/proposed IFC entity
- EDP4. Modified IFC attribute on existing IFC entity
- EDP5. New predefined type container connected to existing/proposed IFC entity.
- EDP6. New predefined type on existing/proposed predefined type container
- EDP7. New IFC property set and/or IFC quantity set
- EDP8. Modelling of IFC select

These patterns provide simple methods and depictions of how to stay within the modelling guidelines and enable the automatic generation of encodings.

5.2.1 EDP1 - New IFC entity and/or entity type as a subtype

IFC entities are the primary classes within IFC. On occasion we need to add new IFC entities to cover new/missing semantics. When new IFC entities are added the addition of a typing entity may be required and should be subtyped in the relevant hierarchy. Figure XX provides an example of the addition of the *IfcCourse* and *IfcCourseType* semantic elements. In addition, it may be required to define predefined types against these new semantic elements the pattern for the definition is elaborated in EDP5, the procedure is the same for adding PDT's to proposed or existing entities.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the new IFC entity(ies) or create the relevant diagram required.
2. Insert the new IFC entity(ies) and provide the NAME and DESCRIPTION in line with modelling guidelines, please follow relevant status pattern to manage development cycle.
3. Locate the relevant supertypes (parent entities) and drag/drop or import into diagram, define the required UML generalisation for inheritance, drawing from source (child entity) to target (parent/supertype entity).
4. **OPTIONAL** - Define the relevant attributes on the IFC entity(ies), using the relevant design pattern (EDP3)
5. **OPTIONAL** – Define the set of predefined types of the entity set, using the relevant design pattern (EDP5)

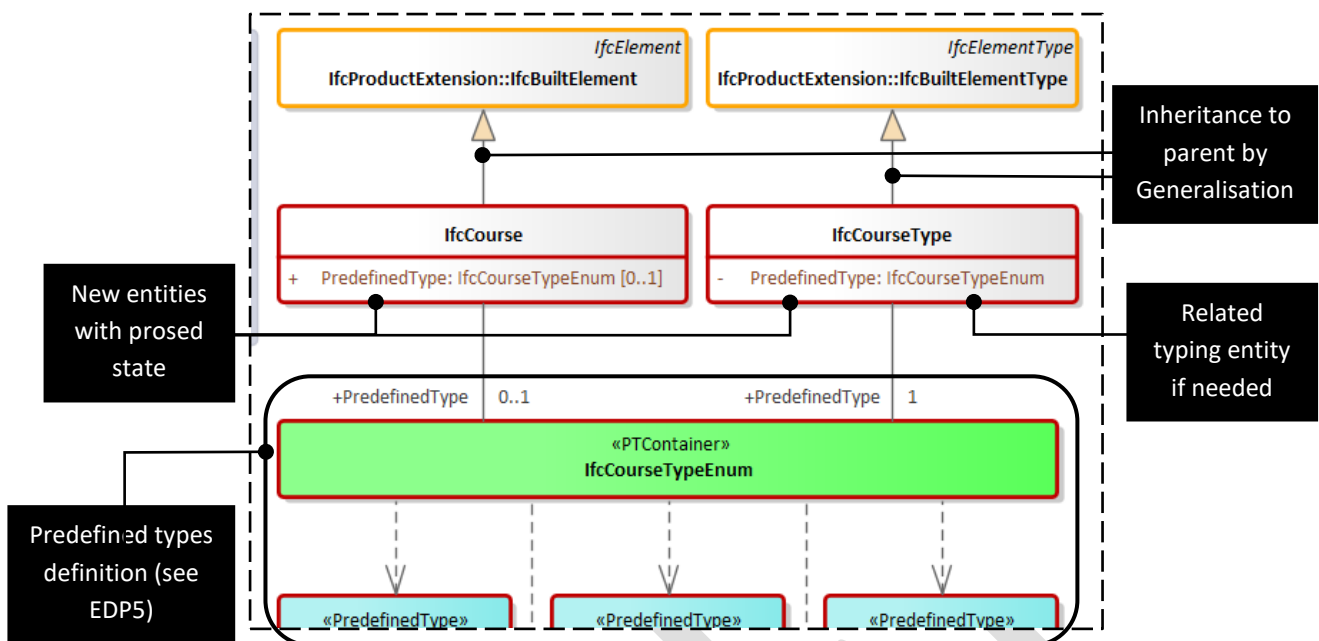


Figure 10 Example of proposed IFC entities and relationships

Limitations of EDP1

- High implementation cost should be avoided if suitable element with a new predefined type can be defined. Consider adjustments to semantics definitions to accommodate.

5.2.2 EDP2 - Modification to an existing IFC entity

IFC entities are the primary classes within IFC. In the event we need to widen the scope of an entity's semantics or modify its implementation the following design pattern is advised. Modifications can include updates to the entity's definition, changes to the entity's name (in compliances with backwards compatibility rules), or changes from abstract to instantiable. For all other modifications please see the more specific design pattern. Figure XX shows an example change where *IfcBuildingElement* has been changed to *IfcBuiltElement*, its definition has been updated to reflect the name and semantic change and lastly it has been changed from abstract to instantiable. To mark changes to description red mark-up is used in the text.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the modifications to the IFC entity(ies) or create the relevant diagram required.
2. Drag/drop or import the IFC entity(ies) into the diagram
3. Make the relevant name, description or abstract setting changes via the development environment. Make sure to stay within the relevant modelling guidelines.
4. Update the status of the IFC entity(ies) per status design patterns.

5. Make sure the proposed modification is conveyed in the relevant XMI export and report generation.

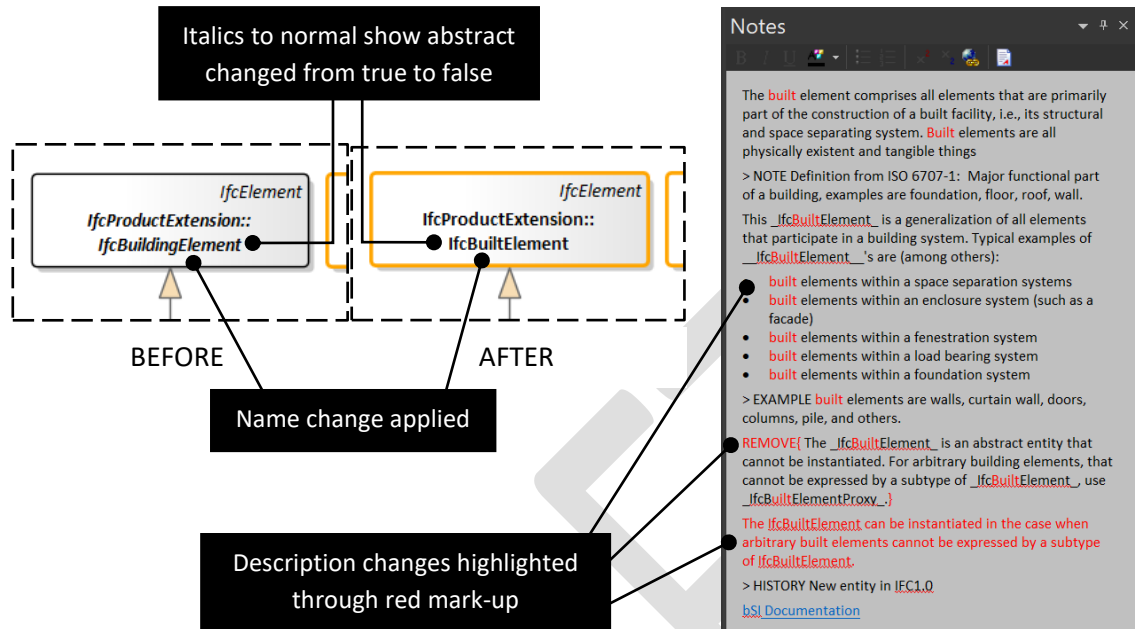


Figure 11 Depiction of IFC entity modifications

Limitations of EDP2

- Correct versioning strategy needed to track changes to existing entities.
- Mark-up will be removed when moved from proposed into implemented status (loss of change tracking unless other processes employed).
- **BACKWARDS COMPATABILITY** – any modifications should meet requirements for backwards compatibility this is a vital requirement and consultation.

5.2.3 EDP3 - New IFC attribute on existing/proposed IFC entity

New IFC attributes can be applied to existing or proposed IFC entities, the procedure is the same in either case. With the only variation being the application of the proposed modification status on existing IFC entities or proposed status on new IFC entities. The following design pattern provides an example of 2 new attributes applied to the existing IFC entity *IfcFacilityPart*.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the modifications to the IFC entity(ies) or create the relevant diagram required.
2. Drag/drop or import the IFC entity(ies) into the diagram
3. Add the new IFC attribute(s) to the IFC entity in question with scope of PUBLIC and the relevant type declared in accordance with the modelling guidelines.

4. Update the status of the IFC entity(ies) per status design patterns.
5. Create the related UML association between the source entity and the defined type of the attribute.
6. Set the source and target properties on the UML association and the required multiplicities.

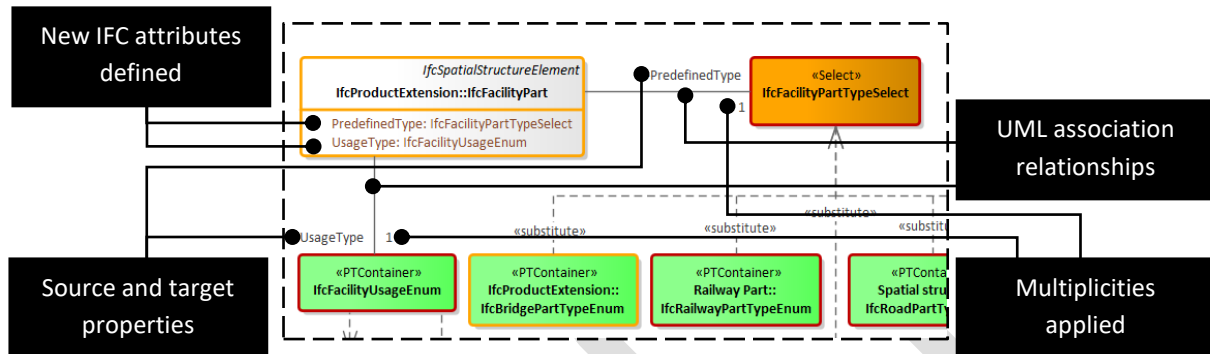


Figure 12 - Depiction of new attribute definitions.

Limitations of EDP3

- Verbose definition of attributes and relationships to aid automation.
- No native means to define EXPRESS INVERSE attributes (this is to be developed and described)

5.2.4 EDP4 - Modified IFC attribute on existing IFC entity

Modifications to IFC attributes can be applied to existing IFC entities, unlike new IFC attributes a stereotype is required on the attribute and the UML association to highlight/flag the modification to readers and automation pipelines. The following design pattern provides an example of the modification of 2 attributes/relationships on *IfcRelInterferesElement*.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the modifications to the IFC entity(ies) or create the relevant diagram required.
2. Drag/drop or import the IFC entity(ies) into the diagram
3. Enact the modifications to the IFC attribute(s) on the IFC entity in question in accordance with the modelling guidelines.
4. Update the status of the IFC entity(ies) to **proposed modification** per status design patterns.
5. Apply the <<Modified>> stereotype to the modified attributes.
6. Update the related UML associations if necessary (below is a move to a different concept).
7. Set the source and target properties on the UML association and the required multiplicities.
8. Apply the <<Modified>> stereotype to the modified UML associations

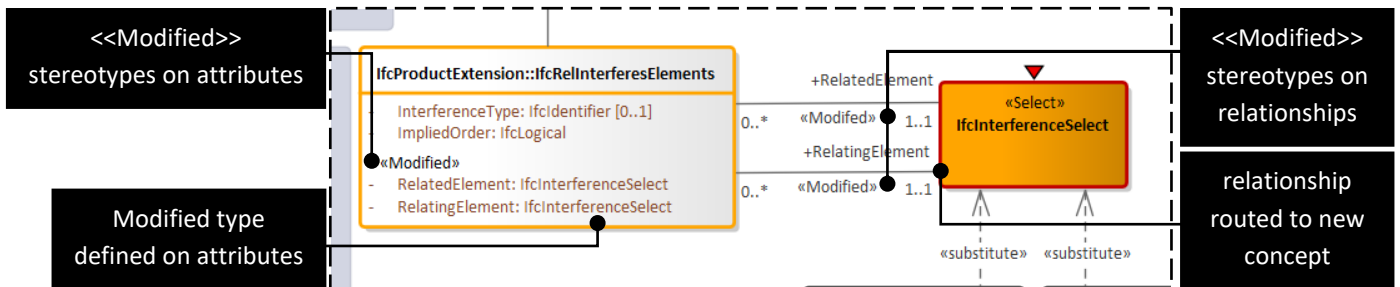


Figure 13 - Depiction of modifications to IFC attributes and/or relationships

Limitations of EDP4

- Verbose definition of attributes and relationships to aid automation.
- No native means to define EXPRESS INVERSE attributes (this is to be developed and described)

5.2.5 EDP5 - New predefined type container connected to existing/proposed IFC entity

Predefined types (PDTs) are an IFC construct that are used to further specialise IFC entities without expanding the implementation load of the EXPRESS schema. PDTs can be proposed on existing (if they don't have any already) and new IFC entities. PDTs are grouped under a PDT container which is represented in EXPRESS as an **ENUMERATION**. PDT containers are connected to IFC entities (and type entities) via a UML association and Attribute name *PredefinedType*. This attribute usually has a 0-1 multiplicity on IFC instance entities and a 1-1 multiplicity on IFC type entities. This design pattern provides an example of the definition of a new predefined type container with relevant UML associations, the container is then populated according to extensions Design Pattern 5 using UML classes and UML dependency relationships.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the new PDT container or create the relevant diagram required.
2. Insert the relevant IFC entity(ies) that will have the new PDT container associated. If IFC entities are new author their relevant information according the design patterns.
3. Add the new predefined type container to the model this is a UML Class with the <<PDTContainer>> stereotype and its name is suffixed with Enum.
4. Author the association between the IFC entity(ies) and the predefined type container with the target property of predefined type and the relevant multiplicities.
5. Define the relevant predefined types according to extension design pattern 5.

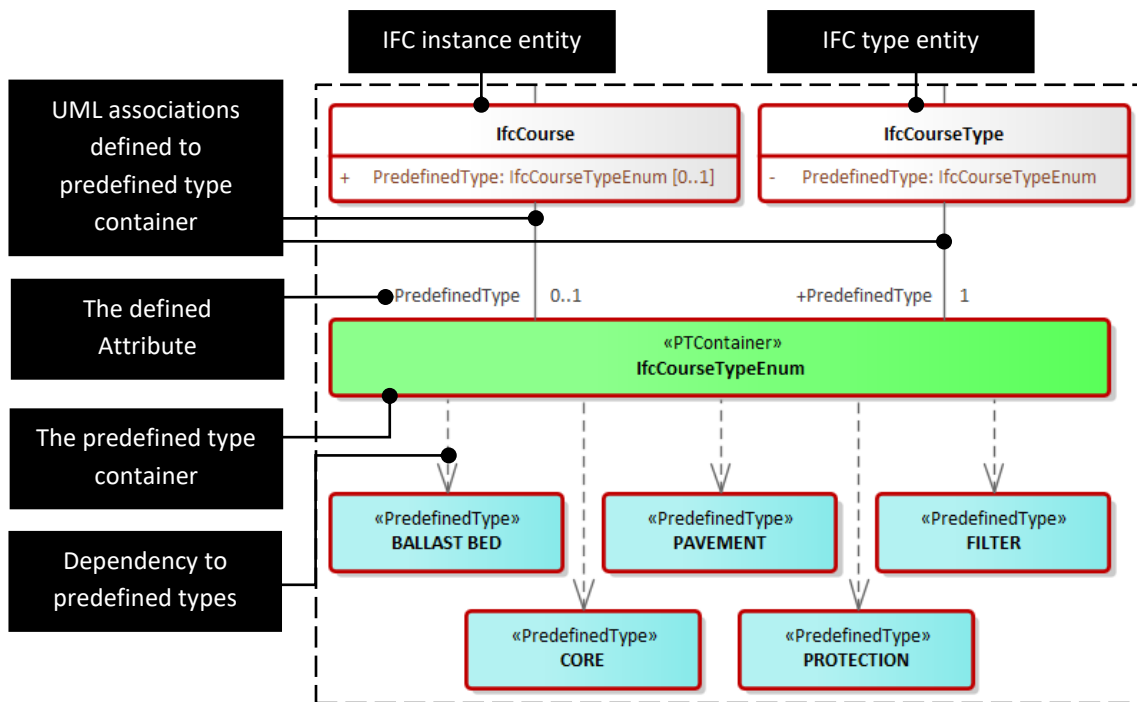


Figure 14 - Depiction of new predefined type containers

Limitations of EDP5

- Verbose definition of predefined types using different structures to the EXPRESS definition, this is to aid readability, semantic definition and association of property sets.

5.2.6 EDP6 – New predefined type on existing/proposed predefined type container

Predefined types (PDTs) are an IFC construct that are used to further specialise IFC entities without expanding the implementation load of the EXPRESS schema. PDTs can be proposed on existing (if they don't have any already) and new IFC entities. PDTs are grouped under a PDT container which is represented in EXPRESS as an **ENUMERATION**. PDTs are related to their containers via a UML dependency. This design pattern provides an example of the definition of a new predefined type within an existing or proposed predefined type container with relevant UML dependencies.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the new predefined type or create the relevant diagram required.
2. Insert the relevant IFC entity(ies) and the PDT container that will contain the new predefined type.
3. Add the new predefined type to the model this is a UML Class with the `<<PredefinedType>>` stereotype and its name authored according the modelling guidelines.

4. Author the UML dependency between the PDT container and the predefined type with arrow pointing towards the predefined type.

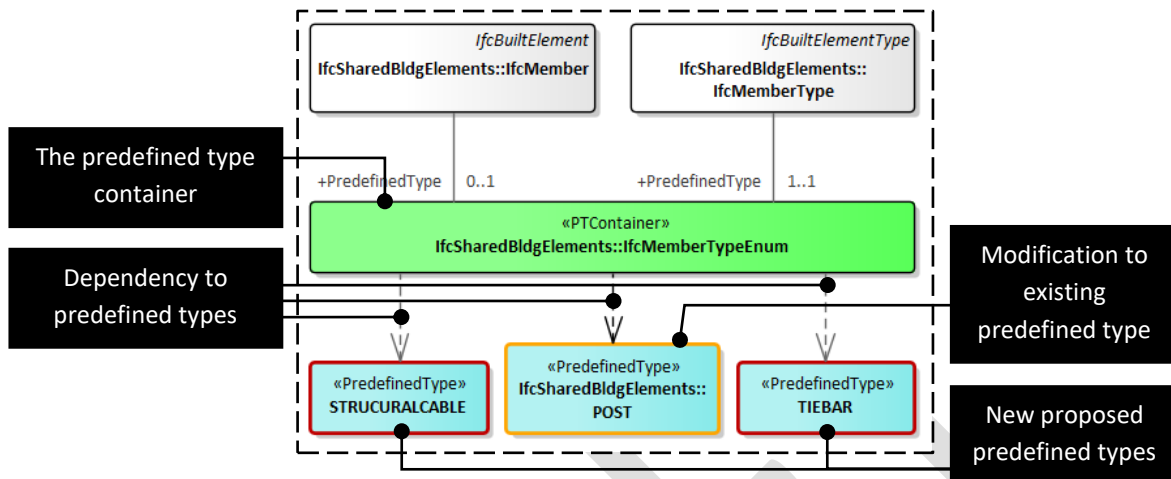


Figure 15 - Depiction of new and modified predefined types

Limitations of EDP6

- Verbose definition of predefined types using different structures to the EXPRESS definition, this is to aid readability, semantic definition and association of property sets.

5.2.7 EDP7 - New IFC property set and/or IFC quantity set

Property sets and quantity sets provide related groups of properties to IFC concepts. This design pattern illustrates the definition of property and quantity sets as UML interfaces and they are then related to the relevant IFC concepts via a UML realization relationship. The attributes/properties contained within the set are defined in the same manner as IFC attributes illustrated by extension design patterns 3 and 4.

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the new property/quantity set or create the relevant diagram required.
2. Insert the relevant IFC concepts that will be associated with the property/quantity set.
3. Add the new property/quantity set to the model, this is a UML Interface with the <<PropertySet>> or <<QuantitySet>> stereotype and its name according to the relevant modelling guidelines.
4. Author the relevant properties within the set according to EDP4 with the exception that no UML Association is required.
5. Relate the property/quantity set to the relevant IFC concepts using a Realisation relationship pointing from the concept to the property set.

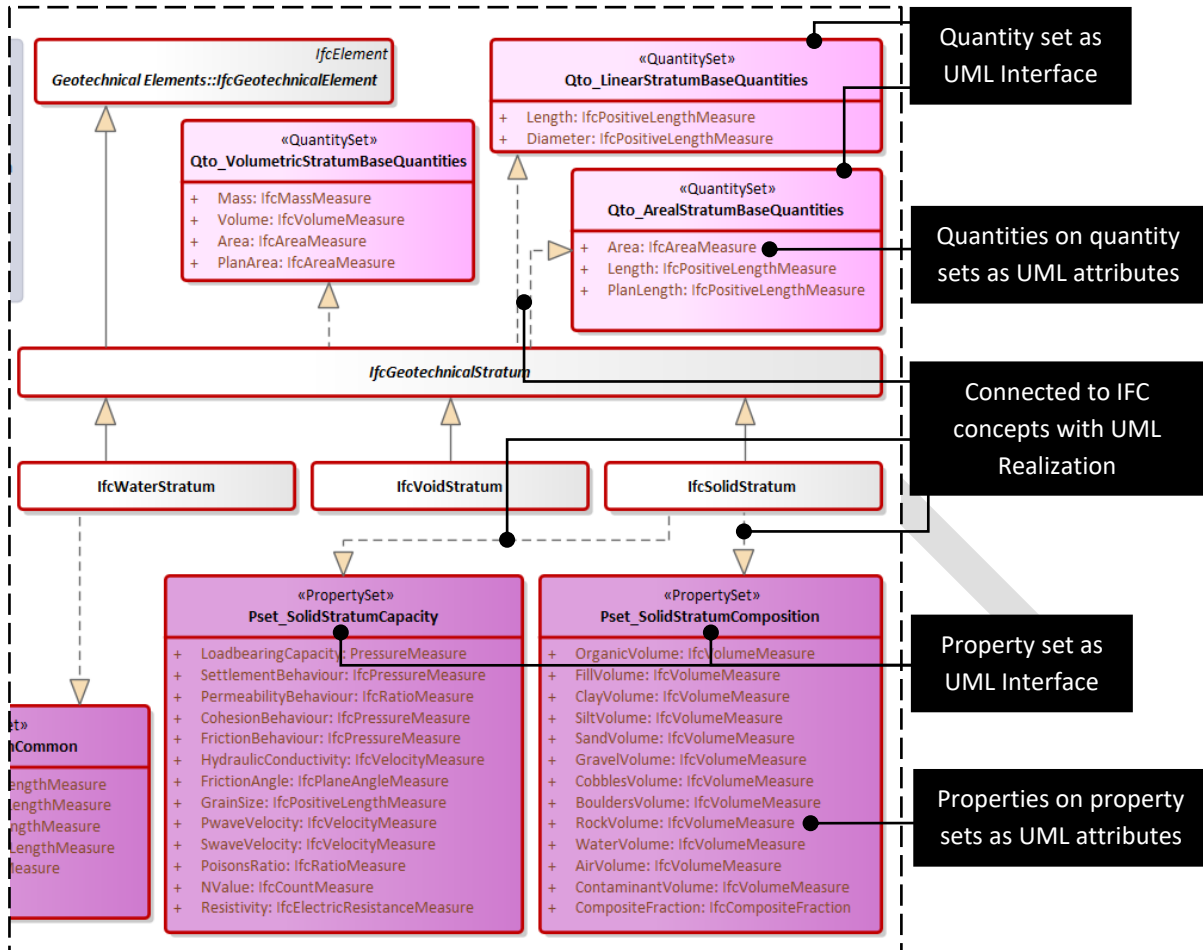


Figure 16 - definition example of property/quantity sets

5.2.8 EDP8 - Modelling of IFC select

Typical Steps

1. Inside the relevant package (in the project's branch), locate the diagram to illustrate the new IFC select or create the relevant diagram required.
2. Insert the relevant IFC entity(ies) that will be associated with the IFC Select. If IFC entities are new author their relevant information according the design patterns.
3. Add the new IFC select to the model this is a UML Interface with the <<Select>> stereotype and its name is suffixed with Select.
4. Author the UML substitutions between the IFC concepts and the IFC Select to illustrate the substitution with the arrow pointing from the concept to the select.

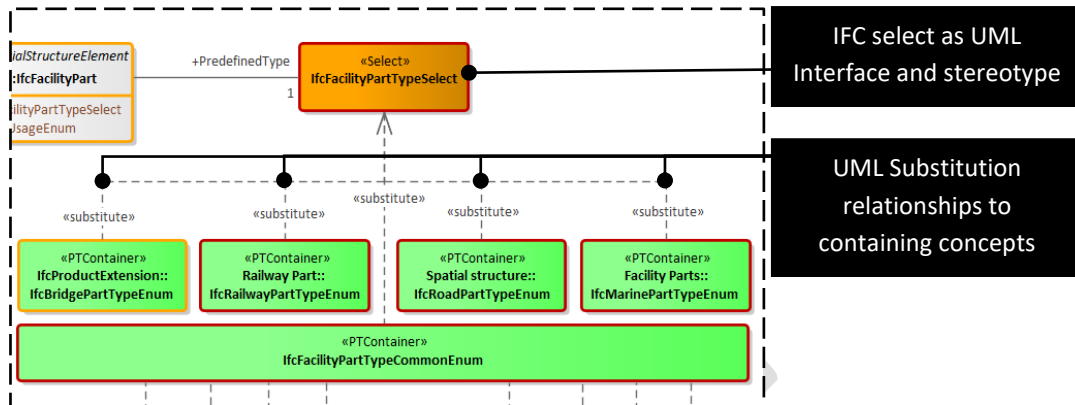


Figure 17 - Modelling of IFC select constructs

5.3 MAPPING DESIGN PATTERNS

IN DEVELOPMENT

DRAFT

6 ANNEXES

DRAFT

Appendix B – IFC Common Schema Contributor List

Company	Name
Stakeholders, bSI InfraRoom Project Steering Committee (IRPSC)	
Austrroads, <i>Australia</i>	Ross Guppy
China Railway BIM alliance, CRBIM , <i>China</i>	Sheng Liming
OYO Corporation, <i>Japan</i>	Hiromasa Shima
The Finnish Transport and Infrastructure Agency, <i>Finland</i>	Tarmo Savolainen
The Swedish Transport Administration, <i>Sweden</i>	Peter Axelsson

Team members	
AEC3, <i>Germany</i>	Thomas Liebich, Matthias Weise
AEC3, <i>UK</i>	Nicholas Nisbet
buildingSMART Australasia, <i>Australia</i>	Jim Plume
CRBIM, <i>China</i>	FeiFei Zhao, Han Bin
Geometry Gym, <i>Australia</i>	Jon Mirtschin
Gobar, <i>The Netherlands</i>	Laura Mol, Fleurike Krijgsman
The Swedish Transport Administration, <i>Sweden</i>	Lars Wikström (Triona), Peter Axelsson
Technical University of Munich, TUM, <i>Germany</i>	André Borrmann
Trimble, <i>Global</i>	Johnny Jensen

Project Organisation and core team

<u>Project leader:</u>	Jim Plume
<u>Technical leader:</u>	Thomas Liebich
<u>Administration:</u>	Laura Mol, Fleurike Krijgsman
<u>Core Project team members:</u>	FeiFei Zhao, Jon Mirtschin, Lars Wikström, Matthias Weise,
<u>IFC Bridge liaison</u>	André Borrmann
<u>IFC Ports & Waterways liaison</u>	Nicholas Nisbet
<u>IFC Rail liaison</u>	Andreas Pinzenöhler, Marion Schenkwein, Peter Axelsson
<u>IFC Road liaison</u>	Karin Anderson
<u>OGC Liaison</u>	Johnny Jensen

Core Harmonisation UML Modelling Team

Alex Bradley (P&W), Chi Zhang (Rail), Evandro Alfieri (Rail), Jon Mirtschin (Geometry Gym), Juha Hyvärinen (Road), Lars Wikstrom (Road), Sergej Muhic (Road), Thomas Liebich (AEC3)

Note: names and companies are simply listed alphabetically

Appendix C – IFC Rail Contributor List

Consortium	Company	Name
bSI	Aec3	Thomas Liebich, Sergej Muhic
	bSI	Aidan Mercer, Jon Proctor, Léon van Berlo, Richard Kelly, Richard Petrie, Sheila Kerai Lum
	PMO	Christian Erismann, Chi Zhang, Dieter Launer, Fei Wang, Guy Pagnier, Winfried Stix (RWR Chairman)
	RWR Steering Committee	Adrian Wildenauer, Christophe Castaing, Franz Josef Peer, Ferraro Modestino, Patrick Offroy, Pierre Etienne Gautier, Peter Axelsson, Sheng Liming, Suo Ning, Tarmo Savolainen
CRBIM	Engineering Management Center of China RAILWAY	Li Zhiyi, Liu Yanhong, Sheng Liming, Shen Dongsheng, Suo Ning
	China Academy of Railway Sciences Corporation Limited (CARS)	Bao Liu, Chen Xuejiao, Hao Rui, Lu Wenlong, Niu Hongrui, Qian Jin, Wang Huilin, Wang Chao, Wang Wanqi, Xie Yalong, Ye Yangsheng, Zhao Youming, Zhi Peng, Zhou Li, Zhu Jiansheng
	China Railway Design Corporation (CRDC)	Feng Yan, Kong Guoliang, Li Hualiang, Mao Ning, Qi Chunyu, Su Lin, Wang Changjin, Wu Weifan, Xu lingyan, Yang Xukun, Yao Yiming, Zhang Jian, Zhao Feifei
	China Railway First Survey And Design Institute Group Co.,Ltd.(FSDI)	Gong Yansheng, Hao Shuai, Huang Wenxun, Jin Guang, Li Zhibiao, Qiao Jinxin, Ren Xiaochun, Zhang Xin, Zhao Le
	China Railway SiYuan Survey & Design Group Co., Ltd. (CRFSDI)	Dai Sai, Du Guangyu, Feng Guangdong, Li Yifan, Liu Zhengzi, Liu Lihai, Shen Zhiling, Zhong Qing, Zhou Jieyun, Zhu Dan
	China Railway Eryuan Engineering Group Co. Ltd (CREEC)	Dong Fengxiang, Wang Yong, Wang Huaisong, Wang Xuelin, Yang Gang
FTIA	FTIA	Marion Schenkwein, Tarmo Savolainen, Teea Kantojärvi
MINnD	Egis	Christian Grobost, Christophe Castaing, Mourad Boutros, Vincent Keller
	Railenium	Matthieu Perin, Samir Assaf
	Systra	Louis-marie Borione

ÖBB	IQ soft	Andreas Pinzenöhler
	ÖBB	Alexander Wurm, Attila Szabo, Christoph Burkia, Ewald Griesser, Gerhard Weixler, Martin Neulinger, Richard Mair, Thomas Braatz, Thomas Redl
RFI	Engisis	Evandro Alfieri, Xenia Fiorentini
	RFI	Carpinteri Claudio, Colangiulo Giovanni, Cristofori Enrico, Di Giustino Federica, Domenico Fraioli, Giovanni Sorrentino, Guglielmi Giovanni, Lacomelli Alessio, Lannaioli Marco, Laterza Palma Zaira, Massari Filippo, Rambaldi Ivano
SBB	ETHZ	Odilo Schoch
	RPAG	Marc Pingoud, Claude Marschal, Adonis Engler, Simon Freihart, Patrik Meier, Linus Stauffacher
	SBB	Ali Tatar, Basil Apothéloz, Billal Mahoubi, Cédric Bapst, Daniel Kühni, Grit Meyer, Lukas Schweizer, Marcel Liniger, Rainer Mautz, Raimund Helfenberger, Samlidis Miltiadis
SNCF	SNCF	Achraf Dsoul, Alain Jeanmaire, Cedric Gniewek, Edouard Chabanier, Florian Hulin, Franco Tomassoni, Guillaume Chartier, Heidi Castellanos, Judicael Dehotin, Liliane Bas, Romuald Vernex, Sebastien Buchere, Sondes Karoui, Vincent Thuillier, Vincent Mathouraparsad
Trafikverket	Trafikverket	Lars Wikström, Jitka Hotovcova, Peter Axelsson
TUM	TUM	André Borrmann, Sebastian Esser

Note: names and companies are simply listed alphabetically

Appendix D – IFC Road Contributor List

Company	Name
Stakeholders, bSI InfraRoom Project Steering Committee (IRPSC)	
Apogea, <i>Spain</i>	Jesús Valderrama
APLITOP, <i>Spain</i>	Francisco Navarette
AutoDesk, <i>global</i>	Marek Suchocki
Bundesministerium für Verkehr und digitale Infrastruktur, BMVI , <i>Germany</i>	Gerd Kellermann
China railway BIM alliance, CRBIM , <i>China</i>	Sheng Liming
The Danish Road Directorate, <i>Denmark</i>	Svend Kold Johansen
The Finnish Transport and Infrastructure Agency, <i>Finland</i>	Tarmo Savolainen
Korea Institute of civil engineering and building technology, KICT , <i>Korea</i>	Dr Hyunseok Moon
MIDAS, <i>Korea</i>	Sangyoon Kim
Modélisation des informations interopérables pour les infrastructures durable, MINnD , <i>France</i>	Christophe Castaing
The Norwegian Public Roads Administration, <i>Norway</i>	Hanne Hermanrud
Nye Veier, <i>Norway</i>	Per Qvalben
The Swedish Transport Administration, <i>Sweden</i>	Peter Axelsson
Trimble, <i>Global</i>	Duane Gleason

Company	Name
Team members	
Apogea, <i>Spain</i>	Joaquim Narcis Moya Sala, Antonio Marquez
AEC3	Thomas Liebich
APLITOP, <i>Spain</i>	Salvador Marin, Javier Nadal
AutoDesk, <i>global</i>	Yoshihiko Fukuchi, Heiko Meyerdirks, Tim Yarris, Sebastian Esser (TUM)
BMVI, <i>Germany</i>	Štefan Jaud (TUM), André Borrmann (TUM)
CRBIM, <i>China</i>	Zhao FeiFei, Dongxu Yan, Tianhua Zhu, Hanbin
The Finnish Transport and Infrastructure Agency, <i>Finland</i>	Juho Santala, Jenna Johansson
KICT, <i>Korea</i>	Dr Hyunseok Moon, Jaeyoung Shin, Jisun Won, Xiumei Zheng
MINnD, <i>France</i>	Pauline Gauthier, Nolwenn Lancien,
The Swedish Transport Administration, <i>Sweden</i>	Karin Anderson Lars Wikström (Triona)
Technical University of Munich, TUM, <i>Germany</i>	Štefan Jaud (TUM), André Borrmann (TUM), Sebastian Esser (TUM)
Trimble, <i>Global</i>	Johnny Jensen

Project Organisation and core team

<u>Project leader:</u>	Dr Hyunseok Moon
<u>Co-project leader:</u>	Karin Anderson
<u>Administration:</u>	Andrew Sheil, <i>Ramböll</i> , Laura Vaessen-Mol, <i>Gobar</i>
<u>Technical lead:</u>	Juha Hyvärinen, Jhy OY
<u>IFC lead:</u>	Sergej Muhič, Siemens
<u>Conceptual model lead:</u>	Lars Wikström (Triona) STA
<u>Validation lead:</u>	Štefan Jaud (TUM) BMVI
<u>Property lead/OGC Liaison:</u>	Johnny Jensen, Trimble

WP5, Prototypical implementation, Participating software vendors

Aplitop, Autodesk, Bentley, KICT / Midas IT, TUM, Autodesk, TUM, Obermeyer/ProVI, 12D, Trimble, AKG, Tool, Card-1, CGS-Labs, Istram, SierraSoft, Catenda

Note: names and companies are simply listed alphabetically