



National BIM Standard - United States® Version 3

4 Information Exchange Standards

4.7. Electrical information exchange (SPARKie) - Edition 2013

CONTENTS

4.7.1.1 Business case description	6
4.7.1.1.1 Life-cycle phase list	6
4.7.1.1.2 Business case description	6
4.7.1.1.3 Business case analysis	6
4.7.1.2 Participants and stakeholders	6
4.7.1.2.1 Participants List	6
4.7.1.2.3 Stakeholders coverage analysis	7
4.7.2 Normative references	7
4.7.2.1 References and standards	7
4.7.2.1.1 Reference standards list	7
4.7.2.1.2 Reference standards list (other)	7
4.7.2.1.3 Reference program and project list	7
4.7.3 Terms, definitions, symbols and abbreviated terms	7
4.7.4 Business process documentation	12
4.7.4.1 Process models provided	12
4.7.4.1.1 Business process list	12

4.7.4.1.2 Business process descriptions	13
4.7.4.2 Representative process models	18
4.7.4.2.1 Stakeholder coverage analysis.....	18
4.7.4.2.2 Process coverage analysis.....	20
4.7.4.2.3 Contract documentary deliverable list	20
4.7.4.2.4 Contract documentary deliverable analysis.....	20
4.7.4.3 Process models formatting	21
4.7.4.3.1 BPMN usage description	21
4.7.5 Exchange requirements.....	21
4.7.5.1 Exchange requirements legibility	21
4.7.5.1.1 Exchange requirements list	21
4.7.5.1.2 Exchange requirement classification list	21
4.7.5.1.3 Exchange requirement coverage analysis	21
4.7.5.2 Exchange requirements detail	22
4.7.5.2.1 Exchange requirements definition	22
4.7.5.2.2 Business rule list.....	26
4.7.5.2.3 Business rule definition.....	42
4.7.5.3 Exchange requirements reusability	42
4.7.5.3.1 Related business process list	42
4.7.5.3.2 Related exchange requirement list.....	43
4.7.5.3.3 Related exchange requirement reuse analysis	44

4.7.6 Model view definition	45
4.7.6.1 Data definition.....	45
4.7.6.1.1 Data definitions list.....	45
4.7.6.1.2 Data definitions	47
4.7.6.1.3 Data definition reference schema list	206
4.7.6.2 Concept definitions	207
4.7.6.2.1 Concept list	207
4.7.6.2.2 Concept definitions	210
4.7.6.2.3 Concept attributes list	282
4.7.6.2.4 Concept relationship description	299
4.7.6.2.5 Concept requirements applicability.....	301
4.7.6.3 Concept reusability	334
4.7.6.3.1 Concept list	334
4.7.6.3.2 Related existing concept list	335
4.7.6.3.3 (not used).....	337
4.7.6.3.4 Concept business rule list.....	337
4.7.6.3.5 Concept business rule description.....	361
4.7.6.4 Implementation	377
4.7.6.4.1 MVD schema listing	377
4.7.6.4.2 MVD format description	378
4.7.6.4.3 MVD dynamic schema analysis.....	379

4.7.6.4.4 Non-applicable entity exclusion analysis	399
4.7.7 Conformance Testing Procedures	405
4.7.7.1 Format and content requirements	405
4.7.7.1.1 Test rule list	405
4.7.7.1.2. Test rule definition	405
4.7.7.1.3 Test rules formatting	405
4.7.7.1.4 Test rule coverage analysis	405
4.7.7.2 Examples and mapping requirements	406
4.7.7.2.1 Example file list	406
4.7.7.2.2 Example file description	406
4.7.7.2.3 Common BIM file reuse	406
4.7.7.2.4 Implementers' agreements	406
4.7.7.2.5 Transformations/mapping allowed	406
4.7.7.2.6 Transformation/mapping documentation	406
4.7.7.3 Testing tools and procedures	406
4.7.7.3.1 Testing tool list	406
4.7.7.3.2 Testing tool algorithm	406
4.7.7.3.3 Testing tools sample files	411
4.7.7.3.4 Testing tool software availability	411
4.7.8 Implementation resources	411
4.7.8.1 Implementation resources list	411

4.7.8.1.1 Implementation guides	411
4.7.8.2 Implementation resources completeness	411
4.7.8.2.1 Workflow coverage methodology	411
4.7.8.2.2 Workflow coverage analysis	412
4.7.9 Revision Plans	412
4.7.9.1 Revision plans list	412
4.7.9.1.1 Revision management process	412
4.7.9.1.2 Revision management notification	412
4.7.9.2 Proposed revision deployment methods	412
4.7.9.2.1 Revision management process	412
4.7.9.2.2 Revision management notification	412
Annex A	413
Bibliography	414

4.7.1 Scope – General Criteria

4.7.1.1 Business case description

4.7.1.1.1 Life-cycle phase list

- Criteria Definition Phase
- Design Phase
- Coordination Phase

4.7.1.1.2 Business case description

The objectives of this project, which is part of the Life-Cycle Model For Mission Ready Sustainable Facilities (LCM), are to collect and document the requirements for electrical system design from subject matter experts (SMEs) – architects, engineers and specifiers involved in the design, specification and product selection for interior electrical systems – as well as from published documents on ELie (East, 2012c) and exploratory modeling of electrical system components and connections (SPARKie [East, 2012b]) which has not yet been published, in order to create formal specifications that can be directly applied to open standard building information models at the coordinated design (also called construction documents) stage. This ontology is documented in the present report. An additional modeling report uses these requirements to update standard building models using commercial off-the-shelf (COTS) software in order to demonstrate the capabilities and limitations of the development of open BIM models.

[East, E.W., 2012c. Equipment Layout information exchange (ELie). Available at: <http://www.buildingsmartalliance.org/index.php/projects/activeprojects/114> 17 August 2012.]

[East, E.W., 2012b. Electrical System information exchange (Sparkie). Available at: <http://www.buildingsmartalliance.org/index.php/projects/activeprojects/178> 17 August 2012.]

4.7.1.1.3 Business case analysis

This project documents the process flow and data exchange requirements for the design of electrical distribution systems in typical Army facilities. A subsequent part of this project applies the understanding of the process flow and exchange requirements to a “real-world” situation, examining how the information exchanges can be handled in existing Building Information Models (BIM) software, using modified three example Building Information Models (BIM) based on the BuildingSmart Industry Foundation Class (IFC) BIM schema. The example BIMs include a Duplex Apartment, an Office Building and a Medical Clinic, which represent typical low-rise Army facilities of each of these types. Electrical equipment and systems are updated using commercial off-the-shelf (COTS) BIM software and a native common object library developed in this COTS software. Coordinated IFC models are then generated using the COTS IFC export capability.

4.7.1.2 Participants and stakeholders

4.7.1.2.1 Participants List

Bogen, Chris. US Army Corps of Engineers [D]
 Chipman, Tim. Constructivity.com LLC [D]
 Deutsch, Randy AIA. Deutsch Insights Inc. [A]
 East, Edward William. US Army Corps of Engineers [D]
 Fadojutimi, Omobolawa. Kristine Fallon Associates, Inc. [A]
 Fallon, Kristine. Kristine Fallon Associates, Inc. [A]
 Feldman, Robert A. Kristine Fallon Associates, Inc. [D]

Forester, Jim. Newforma, Inc. [E, D]
Kalin, Mark. Kalin Associates. [S]
King, Susan F. Harley Ellis Devereaux [A]
Nelson, Chris. Nelson Electric [E, S]
Nisbet, Nick. AEC3 UK. [D]
Solvik, Kenneth. Data Design Systems [E, D]
Williams, Gregory. Kristine Fallon Associates, Inc.

4.7.1.2.2 Stakeholder list

- architect [a]
- electrical engineer [e]
- electrical specifiers [s]
- software developers [d]

4.7.1.2.3 Stakeholders coverage analysis

Each participant in Clause 1.2.1 has role(s) indicated according to abbreviations defined in Clause 1.2.2.

4.7.2 Normative references

4.7.2.1 References and standards

4.7.2.1.1 Reference standards list

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries

4.7.2.1.2 Reference standards list (other)

None

4.7.2.1.3 Reference program and project list

East, E.W., Danielle Love and Nicholas Nisbet, 2010. A Life-Cycle Model for Contracted Information Exchange. Proceedings of the CIB W78 2010: 27th International Conference –Cairo, Egypt, 16-18 November 2010.

East, E.W., 2012a. Construction Operations Building Information Exchange (COBie), <http://www.wbdg.org/resources/cobie.php> 6 August 2012]

East, E.W., 2007. Construction Operations Building Information Exchange (COBIE), Requirements Definition and Pilot Implementation Standard, ERDC/CERL TR-07-30, US Army Corps of Engineers, http://www.wbdg.org/pdfs/erdc_cerl_tr0730.pdf August 2007

4.7.3 Terms, definitions, symbols and abbreviated terms

For the purpose of this document, the following definitions and abbreviated terms apply.

4.7.3.1**AEC**

Architecture, Engineering, and Construction

4.7.3.2**AEC/FM**

Architecture, Engineering, Construction, and Facilities Management

4.7.3.3**attribute**

unit of information within an entity, defined by a particular type or reference to a particular entity

NOTE There are three kinds of attributes: direct attributes, inverse attributes and derived attributes.

4.7.3.4**BIM**

Building Information Modeling

4.7.3.5**direct attribute**

scalar values or collections including Set (unordered, unique), List (ordered), or Array (ordered, sparse) as defined in [ISO 10303-11]

NOTE Similar to the term "field" in common programming languages.

4.7.3.6**derived attribute**

unit of information computed from other attributes using an expression defined in the schema

4.7.3.7**constraints on attributes**

data type restricting the values of attributes

NOTE 1 The most general constraint is about the existence of attribute values. There are basically two types: mandatory and optional attributes. Values of mandatory attributes must be provided whereas values of optional attributes may be omitted.

NOTE 2 For aggregation data types such as Set, List, or Array, the existence constraint is often refined by a minimal and maximal number of elements, which is also known as cardinality.

4.7.3.8**entity**

class of information defined by common attributes and constraints as defined in [ISO 10303-11]

NOTE Similar to the term "class" in common programming languages but describing data structure only (not behavior such as methods).

4.7.3.9**GUID**

Globally Unique Identifier

4.7.3.10**identification**

capability to find, retrieve, report, change, or delete specific instances without ambiguity

instance

occurrence of an entity

NOTE Similar to the term "instance of a class" in object oriented programming.

4.7.3.11**IFC**

Industry Foundation Classes

4.7.3.12

IFD

International Framework for Dictionaries

4.7.3.13

inverse attribute

unit of information defining queries for obtaining related data and enforcing referential integrity

NOTE Similar to the term "navigation property" in entity-relational programming frameworks.

4.7.3.14

object

anything perceivable or conceivable that has a distinct existence, albeit not material

4.7.3.15

type

basic information construct derived from a primitive, an enumeration, or a select of entities

NOTE 1 Similar to the "Type" construct as defined in [ISO 10303-11].

NOTE 2 Similar in concept to "typedef" or "value type" in common programming languages.

4.7.3.16

select

construct that allows an attribute value to be one of multiple types or entities

NOTE 1 Similar to the "Select" construct as defined in [ISO 10303-11].

NOTE 2 Similar to a "marker interface" in common programming languages.

4.7.3.17

enumeration

construct that allows an attribute value to be one of multiple predefined values identified by name

NOTE 1 Similar to the "Enumeration" construct as defined in [ISO 10303-11].

NOTE 2 Similar in concept to "enum" in common programming languages.

4.7.3.18

actor

person, an organization, or person acting on behalf of an organization

NOTE A specialization of the general term object.

4.7.3.19

classification

categorization, the act of distributing things into classes or categories of the same type

4.7.3.20

constraint

restriction for a specified reason

NOTE A specialization of the general term control.

4.7.3.21

control

directive to meet specified requirements such as for scope, time, and/or cost

NOTE A specialization of the general term object.

4.7.3.22

dictionary

collection of words, terms or concepts, with their definition

4.7.3.23**element**

tangible physical product that can be described by its shape representation, material representations, and other properties

NOTE A specialization of the general term product.

4.7.3.24**element occurrence**

element's position within the project coordinate system and its containment within the spatial structure

4.7.3.25**external reference**

link to information outside the data set, with direct relevance to the specific information the link originates from inside the data set

4.7.3.26**feature**

parametric information and additional property information modifying the shape representation of an element to which it applies

4.7.3.27**group**

collection of information that fulfils a specified purpose

NOTE A specialization of the general term object.

4.7.3.28**library**

catalogue, database or holder of data, that is relevant to information in the data set

NOTE It is information referenced from an external source that is not copied into the data set.

4.7.3.29**object occurrence**

characteristics of an object as an individual

NOTE Similar to "object", "instance", "individual" in other publications.

4.7.3.30**object type**

common characteristics shared by multiple object-occurrences

NOTE Similar to "class", "template", "type" in other publications.

4.7.3.31**process**

object-occurrence located in time, indicating "when"

4.7.3.32**process occurrence**

conceptual object that may occur at a particular time

4.7.3.33**process type**

common characteristics shared by multiple process occurrences

4.7.3.34**product**

physical or conceptual object that occurs in space

NOTE It is specialization of the general term object.

4.7.3.35**product occurrence**

physical or conceptual object that may have a location in space and shape characteristics

4.7.3.36**product type**

common characteristics shared by multiple product occurrences

4.7.3.37**project**

encapsulation of related information for a particular purpose providing context for information contained within

NOTE Context information may include default measurement units or representation context and precision.

4.7.3.38**property**

unit of information that is dynamically defined as a particular entity instance

NOTE Similar to "late-bound" or "run-time" in programming terminology.

4.7.3.39**property occurrence**

unit of information providing a value for a property identified by name

4.7.3.40**property template**

metadata for a property including name, description, and data type

NOTE Similar in concept to "extension property" in common programming languages.

4.7.3.41**property set occurrence**

unit of information containing a set of property occurrences, each having a unique name within the property set

4.7.3.42**property set template**

set of property templates serving a common purpose and having applicability to objects of a particular entity

NOTE Similar in concept to "extension class" in common programming languages.

4.7.3.43**proxy**

object that does not hold a specific object type information

NOTE a specialization of object occurrence.

4.7.3.44**quantity**

measurement of a scope-based metric, specifically length, area, volume, weight, count, or time

4.7.3.45**quantity occurrence**

unit of information providing a value for a quantity

4.7.3.46**quantity set**

unit of information containing a set of quantity occurrences, each having a unique name within the quantity set

4.7.3.47**relationship**

unit of information describing an interaction between items

4.7.3.48**representation**

unit of information describing how an object is displayed, such as physical shape or topology

4.7.3.49**resource**

entity with limited availability such as materials, labor, or equipment

NOTE 1 a specialization of the general term object.

NOTE 2 the "resource definition data schemas" section is unrelated to this concept.

4.7.3.50**resource occurrence**

entity with inherent financial cost, which may be passed onto processes, products, and controls to which it is assigned

4.7.3.51**resource type**

common characteristics shared by multiple resource occurrences

4.7.3.52**space**

area or volume bounded actually or theoretically

NOTE a specialization of the general term product.

4.7.3.53**SPF**

STEP Physical File

4.7.3.54**STEP**

STandard for the Exchange of Product data

4.7.3.55**URI**

Uniform Resource Identifier

4.7.3.56**UUID**

Universally Unique Identifier

4.7.4 Business process documentation**4.7.4.1 Process models provided****4.7.4.1.1 Business process list**

Criteria Phase

Develop Facility Occupancy Model
Document Electrical Project Requirements
Propose Electrical Equipment Room Requirements
Program Spaces, Area, and Budget
Coordinate Development of Concept Design
Select Main Electrical System Types
Develop Electrical Basis of Design
Determine Source of Supply
Propose Electrical Space Requirements
Estimate Energy Performance
Document Concept Design & Estimated Costs

Schematic Design Phase

Locate Electrical Loads
Propose Electrical Equipment Requirements
Propose Electrical Spatial Requirements
Locate and Size Electrical Equipment Room(s)
Propose Lighting Layout
Size Electrical System
Create Raceway Layout
Document Electrical Systems Schematic Design
Coordinate With Other Building Systems
Estimate Energy Usage

Coordinated Design Phase

Update Facility Spatial Configuration
Determine Electrical Supply Requirements
Calculate System Loads
Create Cabling Schematic
Redistribute End User Circuits
Update Cabling and Equipment Size
Select Architectural Light Fixtures
Develop Product Type Specifications/Candidates
Update Facility Spatial Configuration
Document Electrical Design Coordinated
Coordinate With Other Building Systems

4.7.4.1.2 Business process descriptions**Criteria Phase****Develop Facility Occupancy Model**

Architect receives document(s) from the Owner. Architect uses these documents, in conjunction with Building Code guidelines and standards to develop the Facility Occupancy Model.

Document Electrical Project Requirements

Determine system types for consideration. Determine scope of major HVAC equipment. Determine power density at building scope

Propose Electrical Equipment Room Requirements

Verify owner's list of equipment that may impact electrical load and location.

Program Spaces, Area, and Budge

Program spaces according to size and proximity requirements. Verify space sizes and ideal shapes of rooms with electrical service provider, including equipment sizes and clearances around and between equipment.

Coordinate Development of Concept Design

Review and modify spaces and areas per service provider and consultant input.

Select Main Electrical System Types

Determine electrical systems – three-phase vs. single-phase. Determine transformers between systems.

Develop Electrical Basis of Design

Document process model, constraints, formulas, and tables used for making decisions on electrical design, including: lighting calculations showing required and designed foot-candles; estimated panel board loading (including 25% extra as a projection of future building loads); a projection/summation of the panel board loads to justify the sizing of the building transformers; An economic analysis to justify the selection of either 120V/208V or 277Y/480V on the secondary side of the building transformers; an analysis, for the 277Y/480 V choice, as to whether the step down transformer(s) shall be large central units or smaller units placed throughout the building; a short-circuit analysis to determine the AIC rating of the system components; a coordination study to determine the circuit breaker settings and system coordination.

Determine Source of Supply

Perform economic analysis to justify selection of electrical supply source(s).

Propose Electrical Space Requirements

Estimate and verify additional voltage requirements (e.g., annual, seasonal or unusual circumstances), including electrical back-up generators, per code and ordinances. Re-size room accordingly.

Estimate Energy Performance

Estimate and verify additional voltage requirements (e.g., annual, seasonal or unusual circumstances), including electrical back-up generators, per code and ordinances. Re-size room accordingly.

Document Concept Design & Estimated Costs

Calculate construction costs at system level. Calculate operating costs at system level.

Schematic Design Phase**Locate Electrical Loads**

Architect uses the recommendations and preliminary schedule from the Electrical Engineer to indicate locations of major electrical loads in the initial schematic plans.

Propose Electrical Equipment Requirements

Generate One Line Diagram (http://en.wikipedia.org/wiki/One-line_diagram). Determine process for acquiring electrical equipment (e.g., design assist) and verify that the process is acceptable to all participating parties. Determine connected load and demand load for each space. Determine diversity coefficients. Determine circuits. Determine loads at distribution points. Select equipment (or candidates) at each occurrence.

Propose Electrical Spatial Requirements

Electrical Engineer uses the Electrical Equipment List and preliminary architectural plans to develop proposed Electrical Space Requirements.

Locate and Size Electrical Equipment Room(s)

Verify location of service access to site. Determine site lighting loads on system. Reserve space for electrical use. Reserve site areas for electrical utilities.

Propose Lighting Layout

Arrange layout of light fixtures in spaces.

Size Electrical System

Review and verify hourly ratings and code requirements for separations between electrical equipment rooms and adjoining spaces (could potentially impact room areas). Size distribution boards, cables, transformers.

Create Raceway Layout

Layout plan for cables, cores, busbars.

Document Electrical Systems Schematic Design

Create plans from building model. Create schedules from items and attributes.

Coordinate With Other Building Systems

Electrical Engineer sends the electrical drawings to the Architect.

Estimate Energy Usage

Calculate connected loads and demand loads at each circuit, and for overall electrical system. Submit Load Letter (format provided by utility), with values specified per project.

Coordinated Design Phase**Update Facility Spatial Configuration**

Architect revises the facility spatial configuration plans based on the results of the coordination that took place at the end of Design Schematic.

Determine Electrical Supply Requirements

Electrical Engineer uses the Product Type Template and updated plans and other-discipline information to determine total electrical supply requirements. Select from compatible product types for each product occurrence [or if required, select 3 compatible product types that are suitable]. Obtain owner's approval.

Calculate System Loads

Add connected loads and demand loads at each circuit, multiply by coefficients.

Create Cabling Schematic

Layout Raceways, Circuits, Distribution Equipment

Redistribute End User Circuits

For circuits with loads above desired factor, split into separate circuits. For circuits with loads below minimum factor, combine circuits

Update Cabling and Equipment Size

Electrical Engineer updates the schedules of raceways, cables, and equipment sizes.

Select Architectural Light Fixtures

Select suitable type(s) from vendor catalogs

Develop Product Type Specifications/Candidates

Resize to meet capacity, selecting alternate product types that fit requirements.

Update Facility Spatial Configuration

Architect revises the facility spatial configuration plans based on the updated electrical layout and spatial requirements provided by the Electrical Engineer.

Document Electrical Design Coordinated

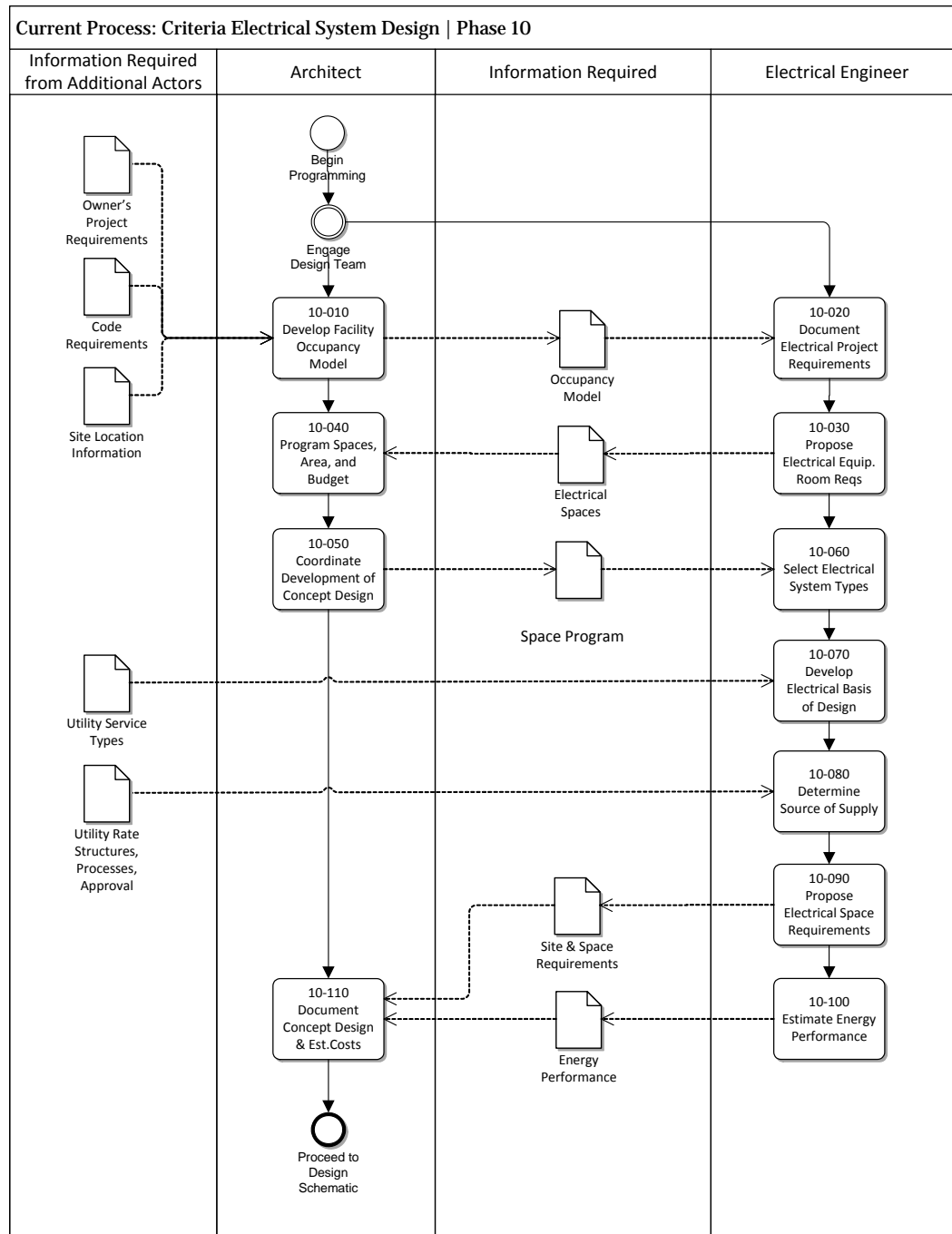
Create plans with detail on elements. Create schedules based on elements and acceptable product types.

Coordinate with Other Building Systems

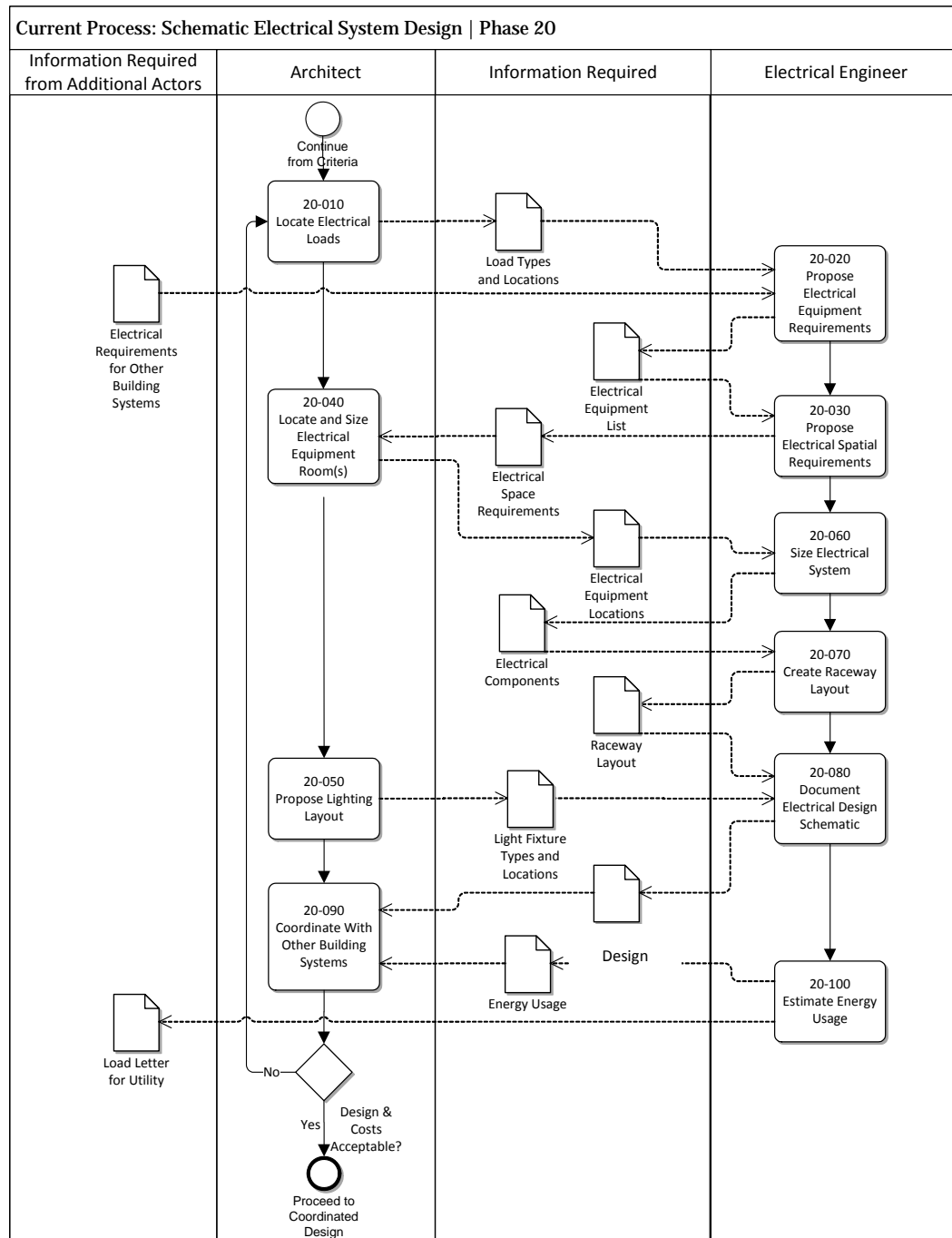
Electrical Engineer sends the electrical drawings to the architect. Architect incorporated the information into the design.

4.7.4.1.3 Business process model diagrams

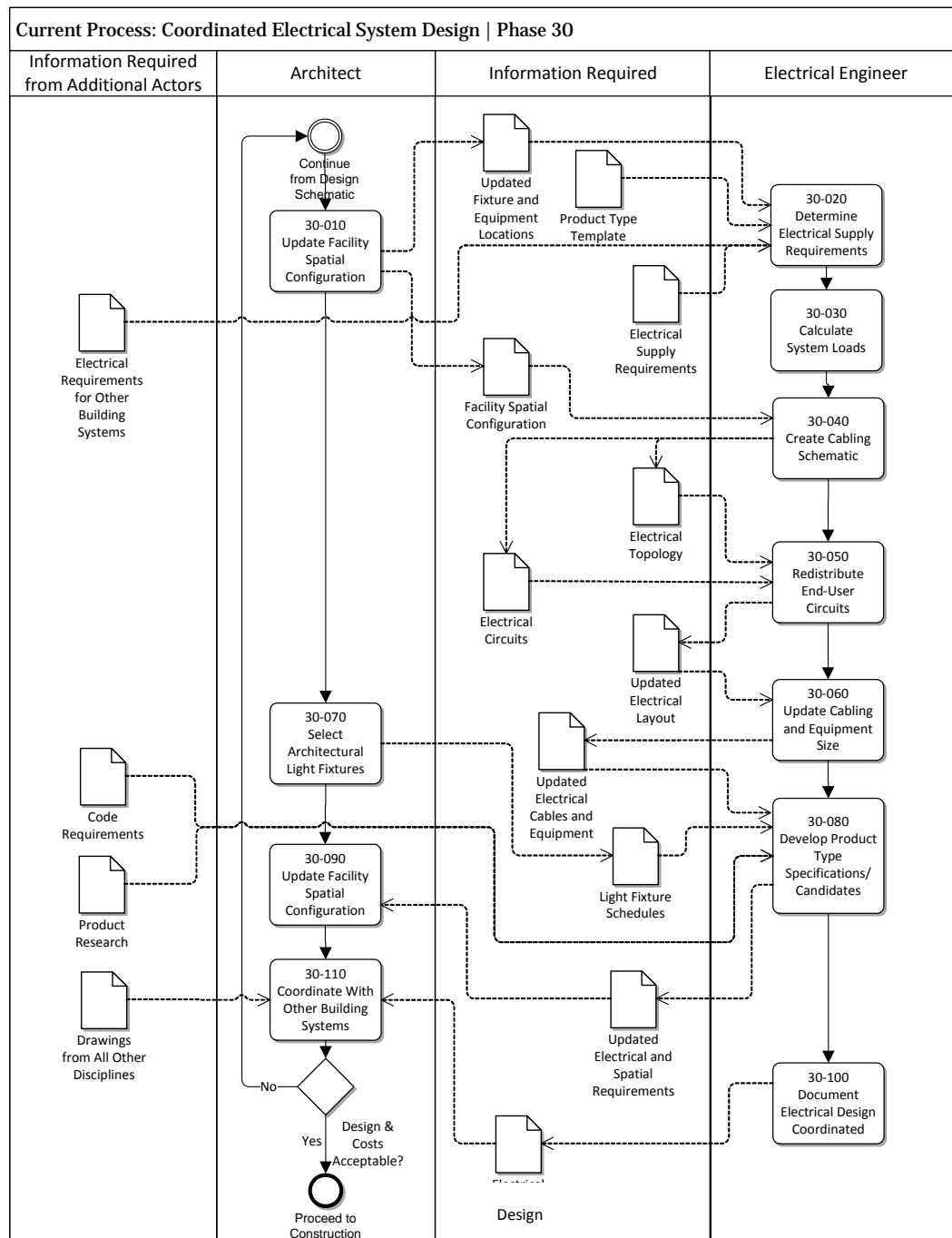
Criteria Phase



Schematic Design Phase



Coordinated Design Phase



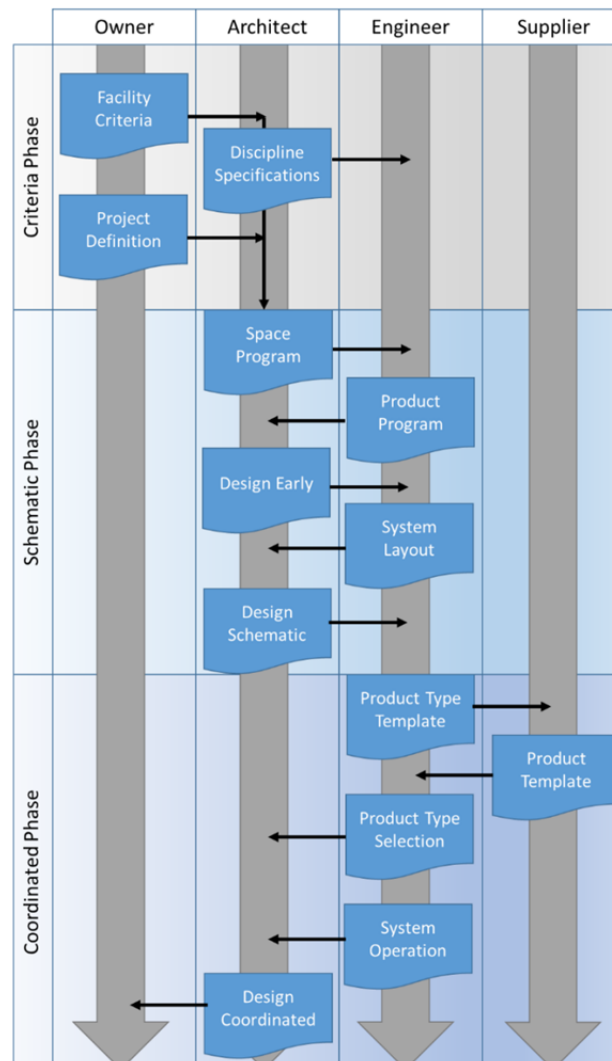
4.7.4.2 Representative process models

4.7.4.2.1 Stakeholder coverage analysis

In order to validate process diagrams and exchange requirements for building electrical distribution systems, the team used the following steps:

- Create “straw-man” process diagrams and task descriptions for each of three phases of the design process (Criteria [also called Programming and Concept Design], Schematic Design [Design Development] and Coordinated Design [Construction Documents]). The process diagrams included suggested steps for the typical Army design process and the task descriptions included suggested information requirements needed to accomplish the task step.
- Assemble a group of subject matter experts (SME) to review and comment on the process diagrams and task descriptions. These experts included two architects, two engineers and two specifiers with experience in the design of building electrical systems.
- Hold meetings with the SME reviewers to explain the process and review criteria.
- Send the process diagrams and task descriptions to the SMEs for their review.
- Analyze the SME comments and contact the SMEs for clarification and additional comments, as needed.
- Revise the process diagrams and task descriptions based on the SME comments.

Processes were rationalized with those in other information exchanges, resulting in the following unified process diagram.



4.7.4.2.2 Process coverage analysis

The team provided “straw-man” process diagrams and task description forms for the design of electrical systems for gathering input from the SME reviewers. The particular selection and sequencing of tasks was intended as a starting point that would be refined using the SME reviewers’ feedback.

4.7.4.2.3 Contract documentary deliverable list

The task forms included the following information:

Item	Description
Task ID	Sequential ID number for the task.
Task Name	A short descriptive name for the task
Information Provider (Roles Involved)	The role or roles that provide the input information necessary to do the task.
Information Provider (Phase)	The stage in the process when the required information is created.
Actor (Roles Involved)	The role or roles that complete the task.
Actor (Phase)	The stage in the process at which the task requires the information.
Information Required	The input information necessary to complete the task.
Current Methods	A short description of the task and its inputs and outputs.

4.7.4.2.4 Contract documentary deliverable analysis

For the task forms, the reviewers were asked:

- Do the task forms accurately and completely detail all information needed to perform the task?
- If not, what is missing?
- Who provides the additional inputs?
- Are Current Methods of performing the task accurately described?

For the process diagrams, the reviewers were asked:

- Although every project has unique circumstances, are the tasks shown in the typically correct order?
- Have we missed any tasks?
- Are there any unnecessary tasks?
- Are all tasks assigned to the correct phase(s)?
- Are all tasks assigned to the correct actor?
- Are all actors that provide the information required indicated?

- Are any extraneous actors indicated?

4.7.4.3 Process models formatting

4.7.4.3.1 BPMN usage description

Models in graphical format are provided in Clause 5.1.3.

4.7.5 Exchange requirements

4.7.5.1 Exchange requirements legibility

4.7.5.1.1 Exchange requirements list

Each exchange is listed by name as follows.

- Facility Criteria
- Discipline Specifications
- Project Definition
- Space Program
- Product Program
- Design Early
- System Layout
- Design Schematic
- Product Type Template
- Product Template
- Product Type Selection
- System Operation
- Design Coordinated

4.7.5.1.2 Exchange requirement classification list

Each phase classification used by this model view is listed by Omniclass notation and title as follows.

- 31-30 00 00 Criteria Definition Phase
- 31-40 00 00 Design Phase
- 31-50 00 00 Coordination Phase

4.7.5.1.3 Exchange requirement coverage analysis

Each exchange is listed by name and corresponding classifications for the process undertaken, the sender of the information, and the receiver of the information.

Exchange	Process	Sender	Receiver
Facility Criteria	31-30 00 00 Criteria Definition Phase	34-10 11 00 Owner	34-20 11 11 Architect
Discipline Specifications	31-30 00 00 Criteria Definition Phase	34-20 11 11 Architect	34-20 11 21 Engineer
Project Definition	31-30 00 00 Criteria Definition Phase	34-10 11 00 Owner	34-20 11 11 Architect
Space Program	31-30 00 00 Criteria Definition Phase	34-20 11 21 Engineer	34-20 11 11 Architect
Product Program	31-40 00 00 Design Phase	34-20 11 21 Engineer	34-20 11 21 Engineer

Exchange	Process	Sender	Receiver
Design Early	31-40 00 00 Design Phase	34-20 11 11 Architect	34-20 11 21 Engineer
System Layout	31-40 00 00 Design Phase	34-20 11 21 Engineer	34-20 11 21 Engineer
Design Schematic	31-40 00 00 Design Phase	34-20 11 21 Engineer	34-20 11 11 Architect
Product Type Template	31-50 00 00 Coordination Phase	34-20 11 21 Engineer	34-35 10 21 Supplier
Product Template	31-50 00 00 Coordination Phase	34-35 10 21 Supplier	34-20 11 21 Engineer
Product Type Selection	31-50 00 00 Coordination Phase	34-20 11 21 Engineer	34-20 11 11 Architect
System Operation	31-50 00 00 Coordination Phase	34-20 11 21 Engineer	34-20 11 11 Architect
Design Coordinated	31-50 00 00 Coordination Phase	34-20 11 11 Architect	34-10 11 00 Owner

4.7.5.2 Exchange requirements detail

4.7.5.2.1 Exchange requirements definition

Each exchange is listed by name and a description of the information contained.

Facility Criteria

This exchange includes high-level criteria specific to the building to be constructed, but without regard for particular disciplines.

The facility occupancy model describes the site location, owner's project requirements, and building requirements.

The site location indicates the geographic location for determining climate information, and the legal address for determining the jurisdiction and applicable building codes.

Discipline Specifications

Electrical project requirements are based on electrical usage criteria as well as equipment determined from usage criteria of other systems such as HVAC and other mechanical systems.

- Occupation: Number of occupants, hours of occupancy, occupancy type
- Cost: Level of finishes
- Architectural: Size of building, number of floors, floor height
- Environment: Heating, cooling, central/unitary
- Illumination: Lighting level, light sources, daylighting, site lighting
- HVAC: Pumps, chillers, fans (power/area for each space)
- Transport: Elevators, escalators
- Control: Automation systems

Project Definition

This exchange includes initial project information to describe a project and its contents. The owner's project requirements consist of a facility type and a set of space types, each indicating occupancy loads, hours of occupancy, design priorities, and climate control requirements.

Space Program

This exchange captures equipment with significant electrical loads. Spaces are programmed according to size and proximity requirements. Equipment sizes and clearances must be provided.

Electrical requirements for each space are elaborated, and each space lists major equipment for consuming, generating, transforming, and storing electricity. Information is required for selecting main electrical system types.

For devices consuming electricity, the following items are needed:

- Load
- Voltage
- System Type

For devices generating electricity (if any), the following items are needed:

- Capacity
- Output Voltage
- System Type

For devices storing electricity (if any), the following items are needed:

- Connected Load
- Uptime

For spatial electrical demand, the following items are needed:

- Lighting power density
- Appliance power density
- Equipment power density

Rate structures may be defined for each system type, indicating time intervals and usage.

Product Program

This exchange enables an architect to revise the facility spatial configuration plans. Required information includes:

- Spatial Elements (Buildings, Levels, Spaces, etc.)
- Building Elements (Walls, Slabs, Doors, Windows, etc.)
- Distribution Elements (Electrical, HVAC, Plumbing, etc.)
- Spatial Zones
- Systems & Circuits
- Connectivity (Space Boundaries, Ports, Connections, Interferences)
- Actors & Assignments

Energy usage is estimated based on major equipment, power densities indicated at spaces, and occupancy schedules.

Design Early

Light fixtures, outlets, and other devices consuming electricity are placed within the building. The quantity and layout of devices is determined by space classification and electrical power density requirements.

A completed concept design contains requirements for all disciplines as follows.

- Architectural: For each space, area and relation to other spaces is indicated along with any exterior space requirements.
- Mechanical: For each distribution element, ventilation, thermal loads, and fuel types are indicated.
- Structural: For each element, static weight is indicated as well as dynamic loads (such as from elevator accelerating).
- Construction: For large equipment, installation methods, sequencing, and paths are indicated.
- Costs: Construction and operation costs are established for each system.

System Layout

For each space, the connected load and demand load is determined along with diversity coefficients. Lighting zones may span within a single space or across multiple spaces, for which any geometrically contained light fixtures are considered part of the zone. Circuits are allocated and loads are determined at distribution points. For each device occurrence, equipment is specified in one of three ways: (a) a specific product model; (b) an arbitrary specification with required properties indicated; or (c) an arbitrary specification with multiple acceptable product models indicated.

One-line diagrams may be derived from this information.

Loads are calculated by rolling up elements, circuits, and systems as follows:

- Element: Load, Time-phased load
- Circuit: Connected Load, Demand Load, Time-phased load
- System: Connected Load, Demand Load, Time-phased load, Diversity Coefficient

Layout of raceways indicates paths and connectivity of each raceway along with allocation of cables within raceways.

For electrical equipment rooms, locations and connectivity of distribution boards and cable carriers are determined.

Sizing electrical systems involves review and verification of hourly ratings and code requirements for separations between electrical equipment rooms and adjoining spaces (could potentially impact room areas). Information in this exchange is used to size distribution boards, cables, and transformers.

This exchange provides detailed information for connectivity and placement of devices cables, including the following:

- Switch: Location, Lighting Load, Controls
- Outlet: Location, Appliance Load
- Cable Segment: Location, Connections, Load, Length, Wiring method (EMT/ENT/MC/Rigid/etc.)
- Switchgear/Panels
- Junction Box: Location (note: may not be at this level of detail)
- Life Safety devices / control panels: Location
- Lighting: Location
- Telecom: Location
- Generating Equipment: controls, transfer switches, interconnects: Location

All products may have defined types indicating Manufacturer, Model, and Specifications. Such types may also have assigned tasks and resources for procurement, where resource types indicate Supplier, Location, and Cost.

Design Schematic

Electrical schematic design includes information needed to calculate electrical performance in its entirety. This information may be used to derive schedules for systems, panelboards, equipment, light fixtures, and feeders.

For coordination with other building systems, plans are created showing equipment locations as well as cable routing and connectivity. Electrical Schedules for equipment, fixtures, feeders, panelboards are derived.

Lighting may be arranged to indicate specific placement of fixtures. Such fixtures may be attached or hung to surfaces, embedded within coverings, or suspended from ceiling grids. The quantity of fixtures may be determined according to the required lighting power density for the space and the power of each fixture.

Product Type Template Requirements

Document process model, constraints, formulas, and tables used for making decisions on electrical design.

- Lighting calculations showing required and designed foot-candles
- Estimated panel board loading (including 25% extra as a projection of future building loads)
- A projection/summation of the panel board loads to justify the sizing of the building transformers
- An economic analysis to justify the selection of either 120V/208V or 277Y/480V on the secondary side of the building transformers
- An analysis, for the 277Y/480 V choice, as to whether the step down transformer(s) shall be large central units or smaller units placed throughout the building
- A short-circuit analysis to determine the AIC rating of the system components.
- A coordination study to determine the circuit breaker settings and system coordination.

EXAMPLE wright.edu

EXAMPLE navy.mil

Product Template

This exchange includes product type information for specific product models provided by manufacturers.

Product Type Selection

For this exchange, the engineer selects specific electrical equipment models (or an approved list from several manufacturers).

An electrical engineer uses the product type templates, updated plans, and other discipline information to determine total electrical supply requirements. For each electrical device, compatible product types are selected for each product occurrence [or if required, multiple compatible product types are selected that are suitable]. The project delivery method may require the owner's approval for final product selection. The total electrical supply requirements are calculated on each circuit according to connected load, demand load, and diversity factor.

System Operation

Energy usage is estimated based on load profiles of major equipment. Estimating energy usage involves calculating connected loads and demand loads at each circuit, and for the overall electrical system. The end result of this calculation may be captured in a Load Letter (format provided by utility), with values specified per project.

Utilities may require the following information for establishing service:

- Load profile at each device consuming electricity
- Generation profile at each device generating electricity
- Service Location
- Total Area
- Conditioned Space Area
- Type of Heat
- Similar Business: Name, Address, Utility Account Number
- Type of Service: Underground, Overhead, Service Change, Relocation, New, Temporary
- Service Characteristics: Size of Load Wires, Sets of Load Wires Per Phase, Load Wire Type (AL/CU), Terminations: Meterbase/C.T.Cabinet/ConnectionBox/Switchgear/Other
- Service Size (amp): 100/150/200/300/400/600/other
- Voltage: 1P3W-120/240, 3P4W-120/240 (<=200 amps), 3P4W-Wye-120/208, 3P4W-Wye-277/480, Other

- Electric Load Excluding Motor Load (kW): Interior Lighting, Exterior Lighting, Electric Cooking, Water Heating, Dryer, Heat Pump, Heat Pump Strip Heat, Computers, Receptacles, Refrigeration, Electric Heat, AC (tons): Data Processing Load Only, Not Including Data Processing
- Electric Motor Load (Except Heating and AC): Phase, Number of Motors, HP, Voltage, Hours of Operation per week
- Estimated Business Operating Time: Hours Per Week, Month Per Year
- Meter Location Desired
- Service Equipment Location Desired

Design Coordinated

The coordinated design contains full detail for all electrical devices and their placement and interaction with other services within the building.

4.7.5.2.2 Business rule list

Each exchange consists of a set of entity data definitions with usage defined according to business rule concepts. An entity describes an object class having one or more attributes, where each attribute may refer to values, collections, or references to other objects. A concept describes usage of object classes, where allowable values and object types are indicated for specific attributes. Each heading that follows refers to an exchange, where each table row corresponds to an entity, each table column corresponds to an exchange, and each cell indicates whether the concept is used for the entity within the exchange.

Facility Criteria

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Properties for Occurrences	Properties for Types	Actor Assignment	Control Assignment	Process Assignment	Spatial Decomposition	Nesting	Organization Role	Calendar	Footprint Geometry	Site Location
IfcProject	R	O	O	O	O	R							R					
IfcTask	R											R						
IfcWorkCalendar	R										R			O		R		
IfcBuilding	R						R	O										
IfcSite	R												R				O	R
IfcSpaceType							R		R									
IfcOccupant	R							R		R					R			

Discipline Specifications

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Properties for Occurrences	Actor Assignment	Spatial Decomposition	Footprint Geometry	Site Location	Resource Cost	Resource Quantity	Resource Cost Rate
IfcProject	R	O	O	O	O	R				R					
IfcBuilding	R						R	O							
IfcBuildingStorey	R														
IfcSite	R									R	O	R			
IfcSpace	R							O							
IfcTransportElement	R							R							
IfcDistributionSystem	R							R							
IfcOccupant	R							R	R						
IfcSubContractResource	R												R	R	
IfcSubContractResourceType	R														R
IfcUnitaryEquipment	R							O							

Project Definition

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Element Decomposition	Spatial Decomposition	Footprint Geometry	Site Location
IfcProject	R	O	O	O	O	R			R		
IfcBuilding	R						R				
IfcBuildingStorey	R							R			
IfcSite	R								R	R	R
IfcSpace	R										

[illegible]

Space Program

	Site Location													
Entity	Clearance Geometry													
	Footprint Geometry											O	R	
	Spatial Containment													
	Spatial Decomposition	R												
	Element Decomposition													
	Properties for Occurrences													
	Classification							R	R					
	Project Context	R												
	Conversion Units													
	Project Declaration													
	Descriptions		O											
	Revision Control		O											
Identity	R													
Entity	IfcProject	R												
	IfcBuilding	R						R	R					
	IfcBuildingStorey	R							R	R				
	IfcSite	R									R		R	
	IfcSpace	R							R	R				
	IfcTransportElement	R							R					
	IfcElectricGenerator	R										R		
	IfcElectricMotor	R							R				R	
	IfcChiller	R							R					
	IfcPump	R							R					
	IfcUnitaryEquipment	R							R					

Product Program

Site Location	
Clearance Geometry	
Footprint Geometry	
Placement	
Port Connectivity	
Spatial Containment	
Ports	
Spatial Decomposition	R
Element Decomposition	
Properties for Occurrences	
Classification	
Project Context	R
Conversion Units	O
Project Declaration	O
Descriptions	O
Revision Control	O
Identity	R
Entity	
	IfcProject

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Properties for Occurrences	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Footprint Geometry	Clearance Geometry	Site Location
IfcBuilding	R						R										
IfcBuildingStorey	R								R								
IfcSite	R									R					R		R
IfcSpace	R						R							R			
IfcTransportElement	R							R			R	O					
IfcDistributionPort	R							R					R				
IfcCableFitting	R																
IfcElectricAppliance	R							R			R	O					
IfcElectricDistributionBoard	R							R			R	O		O			
IfcElectricFlowStorageDevice	R							R			R	O					
IfcElectricGenerator	R							R			R	O				O	
IfcElectricMotor	R							R			R	O				O	
IfcLightFixture	R							R			R	O					
IfcOutlet	R							R			R	O					
IfcProtectiveDevice	R							R			R	O					
IfcSwitchingDevice	R							R			R	O					
IfcTransformer	R							R			R	O				O	
IfcChiller	R							R			R	O					
IfcFlowMeter	R							R			R	O					
IfcPump	R							R			R	O					
IfcUnitaryEquipment	R							R			R	O					

Design Early

Entity	Site Location	Clearance Geometry	Body Geometry	Footprint Geometry	Axis Geometry	Placement	Spatial Containment	Ports	Spatial Decomposition	Element Decomposition	Aggregation	Properties for Occurrences	Classification	Project Context	Conversion Units	Project Declaration	Descriptions	Revision Control	Identity
IfcProject									R					R	O	O	O	O	R
IfcBuilding													R						R
IfcBuildingStorey									R										R
IfcSite									R										R
IfcSpace							R						R						R
IfcTransportElement												R							R
IfcSlabElementedCase						R	R												R
IfcWallElementedCase						R	R												R
IfcDistributionPort												R							R
IfcDistributionSystem											R								R
IfcElectricDistributionBoard																			R
IfcElectricFlowStorageDevice																			R
IfcElectricGenerator								R	R			R							R
IfcElectricMotor																			R
IfcLightFixture							R	R	R			R							R
IfcProtectiveDevice																			R
IfcSwitchingDevice							R												R
IfcTransformer								R	O			R							R
IfcChiller																			R
IfcFlowMeter																			R
IfcPump												R							R
IfcUnitaryEquipment																			R

System Layout

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Material Profile Set Usage	Properties for Occurrences	Group Assignment	Aggregation	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Body Geometry	Clearance Geometry	Site Location
IfcProject	R	O	O	O	O	R							R									
IfcBuilding	R						R															
IfcBuildingStorey	R											R										
IfcSite	R												R					R	R			R
IfcSpace	R						R										R					
IfcTransportElement	R								R													
IfcSlabElementedCase	R																R		R			
IfcWallElementedCase	R																R	R				
IfcDistributionCircuit	R									R												
IfcDistributionPort	R								R							R						
IfcDistributionSystem	R										R											
IfcCableCarrierFitting	R													R			R					
IfcCableCarrierSegment	R							O	R					R	O		R	R				
IfcCableFitting	R													R								
IfcCableSegment	R																					
IfcElectricAppliance	R													R								
IfcElectricDistributionBoard	R								R					R	R							

IfcElectricFlowStorageDevice	R																			
IfcElectricGenerator	R							R					R	R				O		R
IfcElectricMotor	R							R												
IfcJunctionBox	R																			
IfcLightFixture	R							R					R	R						
IfcOutlet	R							R					R	R						
IfcProtectiveDevice	R							R												
IfcSwitchingDevice	R							R					R	R						
IfcTransformer	R							R					R	O						
IfcChiller	R							R												
IfcFlowMeter	R							R												
IfcPump	R							R												
IfcUnitaryEquipment	R							R												

Design Schematic

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Properties for Occurrences	Group Assignment	Aggregation	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Body Geometry	Clearance Geometry	Site Location	Grid
IfcProject	R	O	O	O	O	R						R										
IfcBuilding	R						R															
IfcBuildingStorey	R										R											
IfcGrid	R															R						R
IfcSite	R											R						R	R		R	
IfcSpace	R						R									R						
IfcTransportElement	R																					
IfcCovering	R															R		R				
IfcSlabElementedCase	R															R	R					
IfcWallElementedCase	R															R	R					
IfcDistributionCircuit	R								R													
IfcDistributionPort	R							R							R	R						

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Properties for Occurrences	Group Assignment	Aggregation	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Body Geometry	Clearance Geometry	Site Location	Grid
IfcDistributionSystem	R							R		R												
IfcCableCarrierFitting	R																					
IfcCableCarrierSegment	R							R					R	R		R	R					
IfcCableFitting	R							R					R	R		R						
IfcCableSegment	R							R					R	R		R	R					
IfcElectricDistributionBoard	R							R					R	R		R						
IfcElectricFlowStorageDevice	R							R					R	R		R						
IfcElectricGenerator	R							R					R	R				O		R		
IfcElectricMotor	R							R					R	R				O				
IfcJunctionBox	R							R					R	R		R						
IfcLightFixture	R							R					R	R		R		O				
IfcOutlet	R							R					R	R		R						
IfcProtectiveDevice	R							R					R	R		R						
IfcSolarDevice	R							R					R	R		R						
IfcSwitchingDevice	R							R					R	R		R						
IfcTransformer	R							R					R	R		R		O				
IfcChiller	R							R					R	R		R		O		R		
IfcFlowMeter	R							R					R	R		R		O		R		
IfcPump	R							R					R	R		R		O		R		
IfcUnitaryEquipment	R							R					R	R		R		O		R		

Product Type Template

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Properties for Types	Type-Based Ports
IfcProject	R	O	O	R	O	R		
IfcTransportElementType	R						R	R
IfcDistributionPort	R							
IfcCableCarrierFittingType	R						R	R
IfcCableCarrierSegmentType	R						R	R
IfcCableSegmentType	R						R	R
IfcElectricApplianceType	R						R	R
IfcElectricDistributionBoardType	R						R	R
IfcElectricFlowStorageDeviceType	R						R	R
IfcElectricGeneratorType	R						R	R
IfcElectricMotorType	R						R	R
IfcJunctionBoxType	R						R	R
IfcLampType	R						R	R
IfcLightFixtureType	R							R
IfcOutletType	R						R	R
IfcProtectiveDeviceType	R						R	R
IfcSolarDeviceType	R						R	R
IfcSwitchingDeviceType	R						R	R
IfcTransformerType	R						R	R
IfcChillerType	R						R	R
IfcFlowMeterType	R						R	R
IfcPumpType	R						R	R
IfcUnitaryEquipmentType	R						R	R

Product Template

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Properties for Types	Type-Based Ports
IfcProject	R	O	O	R	O	R		
IfcTransportElementType	R						R	R
IfcDistributionPort	R							
IfcCableCarrierFittingType	R						R	R
IfcCableCarrierSegmentType	R						R	R
IfcCableSegmentType	R						R	R
IfcElectricApplianceType	R						R	R
IfcElectricDistributionBoardType	R						R	R
IfcElectricFlowStorageDeviceType	R						R	R
IfcElectricGeneratorType	R						R	R
IfcElectricMotorType	R						R	R
IfcJunctionBoxType	R						R	R
IfcLampType	R						R	R
IfcOutletType	R						R	R
IfcProtectiveDeviceType	R						R	R
IfcSolarDeviceType	R						R	R
IfcSwitchingDeviceType	R						R	R
IfcTransformerType	R						R	R
IfcChillerType	R						R	R
IfcFlowMeterType	R						R	R
IfcPumpType	R						R	R
IfcUnitaryEquipmentType	R						R	R

Product Type Selection

	Grid	Site Location	Lighting Geometry	Clearance Geometry	Body Geometry	Footprint Geometry	Axis Geometry	Placement	Port Connectivity	Spatial Containment	Type-Based Ports	Ports	Spatial Decomposition	Element Decomposition	Product Type	Group Assignment	Properties for Types	Properties for	Object Typing	Material Profile Set	Classification	Project Context	Conversion Units	Project Declaration	Descriptions	Revision Control	Identity	Entity
IfcProject													R									R	O	O	O	O	R	
IfcBuilding								R												R							R	
IfcBuildingStorey													R														R	
IfcGrid	R							R																			R	
IfcSite		R			R	R		R					R														R	
IfcSpace								R												R							R	
IfcTransportElement					R			R																			R	
IfcTransportElementType										R					O		R										R	
IfcSlabElementedCase							R																				R	
IfcWallElementedCase								R	R																		R	
IfcDistributionCircuit																R											R	
IfcDistributionPort								R	R									R									R	
IfcDistributionSystem																											R	
IfcCableCarrierFitting																			R	R							R	
IfcCableCarrierFittingType											R				O		R										R	
IfcCableCarrierSegment								R	R																		R	
IfcCableCarrierSegmentType											R				O		R			R							R	
IfcCableFitting								R											R	R							R	
IfcCableSegment								R	R																		R	
IfcCableSegmentType											R				O		R										R	
IfcElectricAppliance																			R	R							R	
IfcElectricApplianceType															O		R										R	
IfcElectricDistributionBoard																			R	R							R	
IfcElectricDistributionBoardType															O		R										R	
IfcElectricFlowStorageDevice												R							R	R							R	
IfcElectricFlowStorageDeviceType															O		R										R	
IfcElectricGenerator												R							R	R							R	

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Material Profile Set	Object Typing	Properties for Types	Group Assignment	Product Type	Element Decomposition	Spatial Decomposition	Ports	Type-Based Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Body Geometry	Clearance Geometry	Lighting Geometry	Site Location	Grid
IfcElectricGeneratorType	R									R		O				R										
IfcElectricMotor	R																		R		R					
IfcElectricMotorType	R									R		O				R										
IfcJunctionBox	R																		R		R					
IfcJunctionBoxType	R									R		O				R										
IfcLamp	R								R	R									R		R					
IfcLampType	R									R		O				R										
IfcLightFixture	R								R	R					R	R	R		R		R		R			
IfcLightFixtureType	R									R		O														
IfcOutlet	R								R	R					R	R	R		R		R					
IfcOutletType	R									R		O				R										
IfcProtectiveDevice	R								R										R		R					
IfcProtectiveDeviceType	R									R		O				R										
IfcSolarDevice	R								R	R									R		R					
IfcSolarDeviceType	R									R		O				R										
IfcSwitchingDevice	R								R	R					R	R	R		R		R					
IfcSwitchingDeviceType	R									R		O				R										
IfcTransformer	R								R	R					R	R	R		R		R					
IfcTransformerType	R									R		O				R										
IfcChiller	R								R	R					R	R	R		R		R					
IfcChillerType	R									R		O				R										
IfcFlowMeter	R								R	R					R	R	R		R		R					
IfcFlowMeterType	R									R		O				R										
IfcPump	R								R	R					R	R	R		R		R					
IfcPumpType	R									R		O				R										
IfcUnitaryEquipment	R								R	R					R	R	R		R		R					
IfcUnitaryEquipmentType	R									R		O				R										

System Operation

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Object Typing	Properties for Occurrences	Properties for Performance	Group Assignment	Aggregation	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Body Geometry	Clearance Geometry	Lighting Geometry	Site Location	Grid
IfcProject	R	O	O	O	O	R								R											
IfcPerformanceHistory										R															
IfcBuilding	R						R																		
IfcBuildingStorey	R												R												
IfcGrid	R													R				R							R
IfcSite	R												R							R	R			R	
IfcSpace	R						R											R							
IfcTransportElement	R																				R				
IfcTransportElementType	R																								
IfcSlabElementedCase	R																	R		R					
IfcWallElementedCase	R																	R	R						
IfcDistributionCircuit											R														
IfcDistributionPort	R								R								R								
IfcDistributionSystem	R											R													
IfcCableCarrierFitting	R																				R				
IfcCableCarrierFittingType	R																								
IfcCableCarrierSegment	R																		R		R				
IfcCableCarrierSegmentType	R																								
IfcCableFitting	R																				R				
IfcCableSegment	R																		R		R				
IfcCableSegmentType	R																								
IfcElectricAppliance	R																				R				
IfcElectricApplianceType	R																								
IfcElectricDistributionBoard	R																				R				
IfcElectricDistributionBoardType	R																								
IfcElectricFlowStorageDevice	R																								
IfcElectricFlowStorageDeviceType	R																								

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Object Typing	Properties for Occurrences	Properties for Performance	Group Assignment	Aggregation	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Body Geometry	Clearance Geometry	Lighting Geometry	Site Location	Grid
pe																									
IfcElectricGenerator	R							R	R						R	R					R	R			
IfcElectricGeneratorType	R																								
IfcElectricMotor	R																				R				
IfcElectricMotorType	R																								
IfcJunctionBox	R																				R				
IfcJunctionBoxType	R																								
IfcLamp	R																				R				
IfcLampType	R																								
IfcLightFixture	R							R	R						R	R		R			R		R		
IfcLightFixtureType	R																								
IfcOutlet	R							R	R						R	R					R				
IfcOutletType	R																								
IfcProtectiveDevice	R							R													R				
IfcProtectiveDeviceType	R																								
IfcSolarDevice	R																				R				
IfcSolarDeviceType	R																								
IfcSwitchingDevice	R							R	R						R	R					R				
IfcSwitchingDeviceType	R																								
IfcTransformer	R							R	R						R	R					R				
IfcTransformerType	R																								
IfcChiller	R																				R				
IfcChillerType	R																								
IfcFlowMeter	R																	R			R				
IfcFlowMeterType	R																								
IfcPump	R																				R				
IfcPumpType	R																								

Grid		
Site Location		
Lighting Geometry		
Clearance Geometry		
Body Geometry	R	
Footprint Geometry		
Axis Geometry		
Placement		
Port Connectivity		
Spatial Containment		
Ports		
Spatial Decomposition		
Element Decomposition		
Aggregation		
Group Assignment		
Properties for Performance		
Properties for Occurrences		
Object Typing		
Classification		
Project Context		
Conversion Units		
Project Declaration		
Descriptions		
Revision Control		
Identity	R	
Entity		
IfcUnitaryEquipment	R	
IfcUnitaryEquipmentType	R	

Design Coordinated

Grid			
Building Location			
Site Location		R	
Lighting Geometry			
Clearance Geometry			
Body Geometry			
Surface Geometry			
Footprint Geometry			
Axis Geometry			
Placement			
Port Connectivity			
Spatial Containment			
Ports			
Spatial Decomposition	R		
Element Decomposition			
Aggregation			
Group Assignment			
Properties for Types			
Properties for Occurrences			
Object Typing			
Material Profile Set			
Classification			
Project Context	R		
Conversion Units			
Project Declaration			
Descriptions			
Revision Control			
Identity	R	O	
Entity			
IfcProject	R	O	
IfcBuilding	R		
IfcBuildingStorey	R		
IfcGrid	R		R
IfcSite	R		R
IfcSpace	R		R
IfcTransportElement	R		R
IfcTransportElementType	R		R
IfcSlabElementedCase	R		R
IfcWallElementedCase	R		R
IfcDistributionCircuit	R		
IfcDistributionPort	R		R
IfcDistributionSystem	R		
IfcCableCarrierFitting	R		R
IfcCableCarrierFittingType	R		R

Entity	Identity	Revision Control	Descriptions	Project Declaration	Conversion Units	Project Context	Classification	Material Profile Set	Object Typing	Properties for Occurrences	Properties for Types	Group Assignment	Aggregation	Element Decomposition	Spatial Decomposition	Ports	Spatial Containment	Port Connectivity	Placement	Axis Geometry	Footprint Geometry	Surface Geometry	Body Geometry	Clearance Geometry	Lighting Geometry	Site Location	Building Location	Grid
IfcCableCarrierSegment	R						R		R	R						R	R		R	R		R						
IfcCableCarrierSegmentType	R						R	R			R																	
IfcCableFitting	R						R		R	R						R	R		R			R						
IfcCableSegment	R						R		R	R						R	R		R	R		R						
IfcCableSegmentType	R						R	R			R																	
IfcElectricAppliance	R						R		R	R						R	R		R			R						
IfcElectricApplianceType	R						R				R																	
IfcElectricDistributionBoard	R						R		R	R						R	R		R			R						
IfcElectricDistributionBoardType	R						R				R																	
IfcElectricFlowStorageDevice	R																											
IfcElectricFlowStorageDeviceType	R						R				R																	
IfcElectricGenerator	R						R		R	R						R	R					R	R					
IfcElectricGeneratorType	R						R				R																	
IfcElectricMotor	R						R		R	R						R	R					R	R					
IfcElectricMotorType	R						R				R																	
IfcJunctionBox	R						R		R	R						R	R		R			R						
IfcJunctionBoxType	R						R				R																	
IfcLamp	R						R		R	R						R	R		R			R						
IfcLampType	R						R				R																	
IfcLightFixture	R						R		R	R						R	R		R			R		R				
IfcLightFixtureType	R						R				R																	
IfcOutlet	R						R		R	R						R	R		R			R						
IfcOutletType	R						R				R																	
IfcProtectiveDevice	R						R		R	R						R	R		R			R						
IfcProtectiveDeviceType	R						R				R																	

Grid																											
Building Location																											
Site Location																											
Lighting Geometry																											
Clearance Geometry																											
Body Geometry																											
Surface Geometry																											
Footprint Geometry																											
Axis Geometry																											
Placement																											
Port Connectivity																											
Spatial Containment																											
Ports																											
Spatial Decomposition																											
Element Decomposition																											
Aggregation																											
Group Assignment																											
Properties for Types																											
Properties for Occurrences																											
Object Typing																											
Material Profile Set																											
Classification																											
Project Context																											
Conversion Units																											
Project Declaration																											
Descriptions																											
Revision Control																											
Identity																											
Entity																											
IfcSolarDevice	R						R		R							R	R		R								
IfcSolarDeviceType	R						R			R																	
IfcSwitchingDevice	R						R		R							R	R		R				R				
IfcSwitchingDeviceType	R						R			R																	
IfcTransformer	R						R		R	R						R	R		R				R				
IfcTransformerType	R						R			R																	
IfcChiller	R						R		R	R						R	R		R				R				
IfcChillerType	R						R			R																	
IfcFlowMeter	R						R		R	R						R	R		R				R				
IfcFlowMeterType	R						R			R																	
IfcPump	R						R		R	R						R	R		R				R				
IfcPumpType	R						R			R																	
IfcUnitaryEquipment	R						R		R							R	R		R				R				
IfcUnitaryEquipmentType	R						R			R																	

4.7.5.2.3 Business rule definition

Business rule definitions are all defined as re-usable templates as indicated in Clause 7.3.5.

4.7.5.3 Exchange requirements reusability

Names and classifications of exchanges are intended to be consistent across other model views where applicable, while the content of a particular exchange is intended to be unique.

4.7.5.3.1 Related business process list

Business processes within this model view are correlated with those used in other model views as follows.

Plumbing System Design	Electrical System Design	Mechanical System Design	Building Automation	Building Programming	Facilities Management Handover	Common Use Definitions	Process
X	X			X	X		31-30 00 00 Criteria Definition Phase
X	X	X			X		31-40 00 00 Design Phase
X	X	X					31-50 00 00 Coordination Phase

4.7.5.3.2 Related exchange requirement list

Exchange requirements within this model view are correlated with those used in other model views as follows.

Plumbing System Design	Electrical System Design	Mechanical System Design	Building Automation	Building Programming	Facilities Management Handover	Common Use Definitions	Exchange
X	X				X	X	Facility Criteria
X	X				X	X	Discipline Specifications
X	X				X		Project Definition
X	X	X			X		Space Program
X	X	X			X		Product Program
X	X				X		Design Early
X	X	X	X		X		System Layout
X	X				X		Design Schematic
X	X	X			X		Product Type Template
X	X	X	X	X	X		Product Template

Exchange	Plumbing System Design					
	Electrical System Design					
	Mechanical System Design					
	Building Automation					
	Building Programming					
	Facilities Management Handover					
Common Use Definitions						
Product Type Selection		X				
System Operation		X		X	X	X
Design Coordinated		X			X	X

4.7.5.3.3 Related exchange requirement reuse analysis

Exchange requirements across other model views are correlated as follows.

Exchange	Plumbing System Design								
	Electrical System Design								
	Mechanical System Design								
	Building Automation								
	Building Programming								
	Facilities Management Handover								
	Common Use Definitions								
	Facility Criteria		X	X			X	X	
	Discipline Specifications		X	X			X	X	
	Project Definition		X				X	X	
Space Program		X				X	X	X	
Product Program		X				X	X	X	
Design Early		X					X	X	
Design Schematic		X					X	X	
Design Coordinated		X					X	X	
Design Issue		X							
Product Type Template		X				X	X	X	

Exchange	Common Use Definitions	Facilities Management Handover	Building Programming	Building Automation	Mechanical System Design	Electrical System Design	Plumbing System Design
Product Template		X		X	X	X	X
Bid Issue		X					
Product Type Selection		X				X	X
System Layout		X		X	X	X	X
Product Installation		X		X			
Product Inspection		X					
Construction Issue		X					
Product Type Parts		X					
Product Type Warranty		X					
Product Type Maintenance		X					
System Operation		X		X	X	X	X
Space Condition		X					
Product Parts Replacement		X					
Space Occupancy		X		X			
Space Activity Renovation		X					
Remodel		X					
Expand		X					
Demolish		X					

4.7.6 Model view definition

4.7.6.1 Data definition

4.7.6.1.1 Data definitions list

Each entity data definition is listed by schema and entity name as follows.

- IfcKernel
- IfcActor
- IfcProject
- IfcControlExtension

- IfcPerformanceHistory
- IfcProcessExtension
- IfcTask
- IfcTaskType
- IfcWorkCalendar
- IfcWorkPlan
- IfcProductExtension
- IfcBuilding
- IfcBuildingStorey
- IfcDistributionElement
- IfcGrid
- IfcSite
- IfcSpace
- IfcSpaceType
- IfcTransportElement
- IfcTransportElementType
- IfcSharedBldgElements
- IfcCovering
- IfcSlabElementedCase
- IfcWallElementedCase
- IfcSharedBldgServiceElements
- IfcDistributionCircuit
- IfcDistributionPort
- IfcDistributionSystem
- IfcSharedFacilitiesElements
- IfcOccupant
- IfcSharedMgmtElements
- IfcCostItem
- IfcCostSchedule
- IfcConstructionMgmtDomain
- IfcConstructionEquipmentResource
- IfcSubContractResource
- IfcSubContractResourceType
- IfcElectricalDomain
- IfcCableCarrierFitting
- IfcCableCarrierFittingType
- IfcCableCarrierSegment
- IfcCableCarrierSegmentType
- IfcCableFitting
- IfcCableSegment
- IfcCableSegmentType
- IfcElectricAppliance
- IfcElectricApplianceType
- IfcElectricDistributionBoard
- IfcElectricDistributionBoardType
- IfcElectricFlowStorageDevice
- IfcElectricFlowStorageDeviceType
- IfcElectricGenerator
- IfcElectricGeneratorType
- IfcElectricMotor
- IfcElectricMotorType
- IfcJunctionBox
- IfcJunctionBoxType
- IfcLamp
- IfcLampType
- IfcLightFixture
- IfcLightFixtureType
- IfcOutlet
- IfcOutletType
- IfcProtectiveDevice
- IfcProtectiveDeviceType
- IfcSolarDevice
- IfcSolarDeviceType
- IfcSwitchingDevice
- IfcSwitchingDeviceType
- IfcTransformer
- IfcTransformerType
- IfcHvacDomain
- IfcChiller
- IfcChillerType

- IfcFlowMeter
- IfcFlowMeterType
- IfcPump
- IfcPumpType
- IfcUnitaryEquipment
- IfcUnitaryEquipmentType

4.7.6.1.2 Data definitions

Each entity data definition is described within subsections as follows, with electronic representations provided in EXPRESS and XSD formats.

IfcActor

The *IfcActor* defines all actors or human agents involved in a project during its full life cycle. It facilitates the use of person and organization definitions in the resource part of the IFC object model. This includes name, address, telecommunication addresses, and roles.

[HISTORY](#) [New entity in IFC2.0](#)

EXPRESS Specification:

ENTITY IfcActor

SUPERTYPE OF (IfcOccupant)

SUBTYPE OF (IfcObject);

TheActor : IfcActorSelect;

INVERSE

IsActingUpon : **SET OF** IfcRelAssignsToActor **FOR** RelatingActor;

END_ENTITY;

Attribute Definitions:

TheActor : Information about the actor.

IsActingUpon : Reference to the relationship that associates the actor to an object.

Inheritance Graph:

ENTITY IfcActor

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0: 1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0: 1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0: 1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0: 1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0: 1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcActor

TheActor : IfcActorSelect;

INVERSE

IsActingUpon : **SET OF** IfcRelAssignsToActor **FOR** RelatingActor;

END_ENTITY;

```

<xs:element name="IfcActor" type="ifc:IfcActor" substitutionGroup="ifc:IfcObject" nillable="true"/>
<xs:complexType name="IfcActor">
  <xs:complexContent>
    <xs:extension base="ifc:IfcObject">
      <xs:sequence>
        <xs:element name="TheActor">
          <xs:complexType>
            <xs:group ref="ifc:IfcActorSelect"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

IfcProject

IfcProject indicates the undertaking of some design, engineering, construction, or maintenance activities leading towards a product. The project establishes the context for information to be exchanged or shared, and it may represent a construction project but does not have to. The *IfcProject*'s main purpose in an exchange structure is to provide the root instance and the context for all other information items included.

The context provided by the *IfcProject* includes:

- the default units
- the geometric representation context for exchange structures including shape representations
- the world coordinate system
- the coordinate space dimension
- the precision used within the geometric representations, and
- optionally the indication of the true north relative to the world coordinate system

[HISTORY New entity in IFC1.0](#)

IFC4 CHANGE The attributes *RepresentationContexts* and *UnitsInContext* are made optional and are promoted to supertype *IfcContext*.

[Informal Propositions:](#)

1. There shall only be one project within the exchange context. This is enforced by the global rule *IfcSingleProjectInstance*.

[EXPRESS Specification:](#)

ENTITY IfcProject

SUBTYPE OF (IfcContext);

WHERE

HasName	: EXISTS(SELF\IfcRoot.Name);
CorrectContext	: NOT(EXISTS(SELF\IfcContext.RepresentationContexts)) OR (SIZEOF(QUERY(Temp < * SELF\IfcContext.RepresentationContexts 'IFCREPRESENTATIONRESOURCE.IFCGEOMETRICREPRESENTATIONSUBCONTEXT' IN TYPEOF(Temp)) = 0);
NoDecomposition	: SIZEOF(SELF\IfcObjectDefinition.Decomposes) = 0;
HasOwnerHistory	: EXISTS(SELF\IfcRoot.OwnerHistory);

END_ENTITY;

[Formal Propositions:](#)

HasName	: The <i>Name</i> attribute has to be provided for IfcProject. It is the short name for the project.
CorrectContext	: If a <i>RepresentationContexts</i> relation is provided then there shall be no instance of <i>IfcGeometricRepresentationSubContext</i> directly included in the set of <i>RepresentationContexts</i> .
NoDecomposition	: The IfcProject represents the root of the any decomposition tree. It shall therefore not be used to decompose any other object definition.
HasOwnerHistory	: The <i>OwnerHistory</i> attribute has to be provided for IfcProject. It provides the minimum of owner information for the project data set and the last change action, that applied to the whole data set. NOTE Each individual data item, that derives from IfcRoot may have an individual <i>OwnerHistory</i> . It then overrides the common ownership and change action information provided at the single IfcProject instance in an IFC data set.

IFC4 CHANGE New where rule.

[Inheritance Graph:](#)

ENTITY IfcProject

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcContext

ObjectType : **OPTIONAL** IfcStrippedOptional;
 LongName : **OPTIONAL** IfcStrippedOptional;
 Phase : **OPTIONAL** IfcStrippedOptional;
 RepresentationContexts : **OPTIONAL SET** [1:?] OF IfcRepresentationContext;
 UnitsInContext : **OPTIONAL** IfcUnitAssignment;

INVERSE

IsDefinedBy : **SET** [0:?] OF IfcRelDefinesByProperties **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDeclares **FOR** RelatingContext;

ENTITY IfcProject**END_ENTITY;**

```
<xs:element name="IfcProject" type="ifc:IfcProject" substitutionGroup="ifc:IfcContext" nillable="true"/>
<xs:complexType name="IfcProject">
  <xs:complexContent>
    <xs:extension base="ifc:IfcContext"/>
  </xs:complexContent>
</xs:complexType>
```

IfcPerformanceHistory

IfcPerformanceHistory is used to document the actual performance of an occurrence instance over time. It includes machine-measured data from building automation systems and human-specified data such as task and resource usage. The data may represent actual conditions, predictions, or simulations.

The realtime data tracked by performance history takes the form of property sets where all properties are based on time series. Unlike design-based data at occurrences and types, performance-driven data is time-sensitive and may change in realtime by some measurement device. Data may be captured at irregular intervals such as when values change beyond established thresholds, or at regular intervals of specified duration.

Declaration use definition

IfcPerformanceHistory may be declared within a project using *IfcRelDeclares* where *RelatingContext* refers to the *IfcProject* and *RelatedDefinitions* includes the *IfcPerformanceHistory*. Default units (used for property sets) are indicated by the declaring project. Only top-level objects are declared; nested performance history objects (through *IfcRelNests*) do not participate in such relationship.

HISTORY New entity in IFC2x2.

EXPRESS Specification:

ENTITY IfcPerformanceHistory

SUBTYPE OF (IfcControl);

LifeCyclePhase : IfcLabel;
PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

Attribute Definitions:

LifeCyclePhase : Describes the applicable building life-cycle phase. Typical values should be DESIGNDEVELOPMENT, SCHEMATICDEVELOPMENT, CONSTRUCTIONDOCUMENT, CONSTRUCTION, ASBUILT, COMMISSIONING, OPERATION, etc.

Inheritance Graph:

ENTITY IfcPerformanceHistory

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcControl

Identification : **OPTIONAL** IfcStrippedOptional;

INVERSE

Controls : **SET OF** IfcRelAssignsToControl **FOR** RelatingControl;

ENTITY IfcPerformanceHistory

LifeCyclePhase : IfcLabel;

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcPerformanceHistory" type="ifc:IfcPerformanceHistory" substitutionGroup="ifc:IfcControl"
nillable="true"/>
<xs:complexType name="IfcPerformanceHistory">
  <xs:complexContent>
    <xs:extension base="ifc:IfcControl">
      <xs:attribute name="LifeCyclePhase" type="ifc:IfcLabel" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcTask

An *IfcTask* is an identifiable unit of work to be carried out in a construction project.

A task is typically used to describe an activity for the construction or installation of products, but is not limited to these types. For example it might be used to describe design processes, move operations and other design, construction and operation related activities as well.

Quantities of resources consumed by the task are dealt with by defining the *IfcElementQuantity* for the resource and not at the instance of *IfcTask*.

HISTORY New entity in IFC1.0. Renamed from *IfcWorkTask* in IFC2x.

IFC4 CHANGE Attributes *TaskTime* and *PredefinedType* added. *IfcMove* and *IfcOrderRequest* has been removed in IFC4 and are now represented by *IfcTask*. *IfcRelAssignsTasks* relationship has been removed as well.

Attribute use definition

Each occurrence of *IfcTask* is given a name that is indicative of its content (*IfcRoot.Name*). A textual description of the task may be provided and this may be further elaborated by a narrative long description (*IfcProcess.LongDescription*). A work method may be declared for the method of work used in carrying out a task. A task is identified as being either a milestone task or not. A milestone task is defined by the marker *IsMilestone*. and has no duration. A status and priority for each task may also be set.

Time and duration use definition

Compared to previous IFC releases, basic task time information (scheduled start time, scheduled finish time, duration) is now directly attached to *IfcTask* through the *TaskTime* attribute. Regular tasks are defined through *IfcTaskTime*. Recurring tasks are defined through *IfcTaskTimeRecurring*. In case a regular task is derived from a recurring task both tasks should be linked together through a *IfcRelNests* relationship, where *IfcRelNests.IsNestedBy* points to the recurring task and *IfcRelNests.Nests* points to all regular tasks that have been derived from the recurring task.

Representation of other activities

The use definitions for *IfcTask* have been generalised to represent other activities as well, including activities that had been defined by own entities in previous IFC releases. This includes

- Order actions
- Move operations

IfcTask represents an order that might be carried out by a Helpdesk acting the role of interface for the organization between the facility user and the functional requirement of fulfilling their needs. The actual task represented by the *IfcTask* entity is turning a request into an order and initiating the action that will enable the order to be completed. The *IfcProjectOrder* or one of its subtypes including maintenance work order, is related to the *IfcTask* using *IfcRelAssignsToControl*.

IfcTask can also be used to describe an activity that moves people, groups within an organization or complete organizations together with their associated furniture and equipment from one place to another. It thus replaces the previous IFC entity *IfcMove*. The functionality is represented in *IfcTask* as follows:

- Move from: The place from which actors and their associated equipment are moving. Use *IfcRelAssignsToProcess* where *RelatingProcess* points to the task and *RelatedObjects* holds the location(s) from which to move.
- Move to: The place to which actors and their associated equipment are moving. Use *IfcRelAssignsToProduct* where *RelatedObjects* points to the task(s) and *RelatingProduct* points to the location to which to move.
- Punch list: A list of points concerning a move that require attention. Use *LongDescription* or else identify sub-tasks to track punch list items individually via *IfcRelNests*.

EXPRESS Specification:

ENTITY *IfcTask*

SUBTYPE OF (*IfcProcess*);

Status	: OPTIONAL <i>IfcStrippedOptional</i> ;
WorkMethod	: OPTIONAL <i>IfcStrippedOptional</i> ;
IsMilestone	: <i>IfcStrippedOptional</i> ;
Priority	: OPTIONAL <i>IfcStrippedOptional</i> ;
TaskTime	: OPTIONAL <i>IfcTaskTime</i> ;
PredefinedType	: OPTIONAL <i>IfcStrippedOptional</i> ;

WHERE

HasName	: EXISTS(SELF\IfcRoot.Name);
CorrectPredefinedType	: NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcTaskTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTaskTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcObject.ObjectType));

END_ENTITY;Attribute Definitions:

Status	: Current status of the task. NOTE Particular values for status are not specified, these should be determined and agreed by local usage. Examples of possible status values include 'Not Yet Started', 'Started', 'Completed'.
WorkMethod	: The method of work used in carrying out a task. NOTE This attribute should not be used if the work method is specified for the IfcTaskType
IsMilestone	: Identifies whether a task is a milestone task (=TRUE) or not (= FALSE). NOTE In small project planning applications, a milestone task may be understood to be a task having no duration. As such, it represents a singular point in time.
Priority	: A value that indicates the relative priority of the task (in comparison to the priorities of other tasks).
TaskTime	: Time related information for the task. IFC4 CHANGE Attribute added
PredefinedType	: Identifies the predefined types of a task from which the type required may be set. IFC4 CHANGE Attribute added

Formal Propositions:

HasName	: The Name attribute should be inserted to describe the task name.
CorrectPredefinedType	: The attribute ObjectType must be asserted when the value of PredefinedType is set to USERDEFINED.

Inheritance Graph:**ENTITY** IfcTask**ENTITY** IfcRoot

GlobalId	: IfcGloballyUniqueId;
OwnerHistory	: OPTIONAL IfcOwnerHistory;
Name	: OPTIONAL IfcLabel;
Description	: OPTIONAL IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments	: SET OF IfcRelAssigns FOR RelatedObjects;
Nests	: SET [0:1] OF IfcRelNests FOR RelatedObjects;
IsNestedBy	: SET OF IfcRelNests FOR RelatingObject;
HasContext	: SET [0:1] OF IfcRelDeclares FOR RelatedDefinitions;
IsDecomposedBy	: SET OF IfcRelAggregates FOR RelatingObject;
Decomposes	: SET [0:1] OF IfcRelAggregates FOR RelatedObjects;
HasAssociations	: SET OF IfcRelAssociates FOR RelatedObjects;

ENTITY IfcObject

ObjectType	: OPTIONAL IfcStrippedOptional;
------------	--

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProcess

Identification : **OPTIONAL** IfcStrippedOptional;
 LongDescription : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsPredecessorTo : **SET OF** IfcRelSequence **FOR** RelatingProcess;
 IsSuccessorFrom : **SET OF** IfcRelSequence **FOR** RelatedProcess;
 OperatesOn : **SET OF** IfcRelAssignsToProcess **FOR** RelatingProcess;

ENTITY IfcTask

Status : **OPTIONAL** IfcStrippedOptional;
 WorkMethod : **OPTIONAL** IfcStrippedOptional;
 IsMilestone : IfcStrippedOptional;
 Priority : **OPTIONAL** IfcStrippedOptional;
 TaskTime : **OPTIONAL** IfcTaskTime;
 PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcTask" type="ifc:IfcTask" substitutionGroup="ifc:IfcProcess" nillable="true"/>
<xs:complexType name="IfcTask">
  <xs:complexContent>
    <xs:extension base="ifc:IfcProcess">
      <xs:sequence>
        <xs:element name="TaskTime" type="ifc:IfcTaskTime" nillable="true" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcTaskType

An *IfcTaskType* defines a particular type of task that may be specified for use within a work control.

HISTORY New entity in IFC4

An *IfcTaskType* provides for all forms of types of task that may be specified. It is a reference definition for a unit of work that may be broken down into (a sequence of) subtasks. Please note that a reference definition can not be part of a workflow definition, i.e. *IfcTaskType* instances define the most abstract level of a reference process without dependencies to other reference processes.

Usage of *IfcTaskType* defines the parameters for one or more occurrences of *IfcTask*. Parameters may be specified through property sets that may be enumerated in the *IfcTaskTypeEnum* data type or through explicit attributes of *IfcTaskType*. Task occurrences (*IfcTask* entities) are linked to the task type through the *IfcRelDefinesByType* relationship.

Figure 75 shows the definition of a task type that is part of a task template library. Please note that in this example the task type is further subdivided into tasks that define task times (for example, duration) and/or a task sequence.

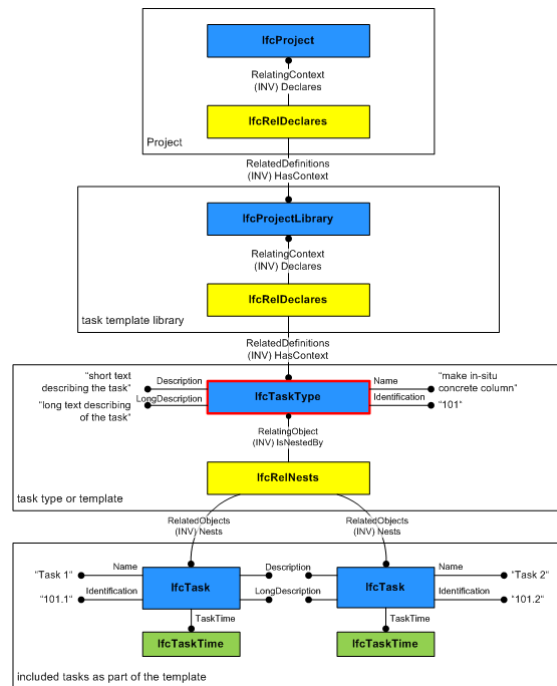


Figure 75 — Task type relationships

EXPRESS Specification:

ENTITY IfcTaskType

SUBTYPE OF (IfcTypeProcess);

PredefinedType : IfcTaskTypeEnum;
WorkMethod : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcTaskTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTaskTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcTypeProcess.ProcessType)) ;

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of a task type from which the type required may be set.
WorkMethod : The method of work used in carrying out a task.

Formal Propositions:

CorrectPredefinedType : The attribute ProcessType must be asserted when the value of PredefinedType is set to USERDEFINED.

Inheritance Graph:**ENTITY** IfcTaskType**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProcess

Identification : **OPTIONAL** IfcStrippedOptional;
 LongDescription : **OPTIONAL** IfcStrippedOptional;
 ProcessType : **OPTIONAL** IfcStrippedOptional;

INVERSE

OperatesOn : **SET OF** IfcRelAssignsToProcess **FOR** RelatingProcess;

ENTITY IfcTaskType

PredefinedType : IfcTaskTypeEnum;

WorkMethod : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcTaskType" type="ifc:IfcTaskType" substitutionGroup="ifc:IfcTypeProcess" nillable="true"/>
<xs:complexType name="IfcTaskType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcTypeProcess">
      <xs:attribute name="PredefinedType" type="ifc:IfcTaskTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcWorkCalendar

An *IfcWorkCalendar* defines working and non-working time periods for tasks and resources. It enables to define both specific time periods, such as from 7:00 till 12:00 on 25th August 2009, as well as repetitive time periods based on frequently used recurrence patterns, such as each Monday from 7:00 till 12:00 between 1st March 2009 and 31st December 2009.

HISTORY New entity in IFC4.

A work calendar is a subtype of *IfcControl* and thus inherits the feature for controlling other objects through *IfcRelAssignsToControl*, which is used to define a work calendar for tasks (*IfcTask*) and resources (*IfcResource*). It also inherits a name and description attribute, whereas a name shall be given and a description may be given as an indication of its content and usage.

The definition of time periods can be derived from a base calendar and/or modified/defined by a set of working times and non-working exception times. All time periods defined by *IfcWorkCalendar.ExceptionTimes* override the time periods inherited from the base calendar (base calendar is defined as the next applicable calendar for the task or resource). Thus, exception times replace the working times from the base calendar.

Figure 87 shows the definition of a work calendar, which is defined by a set of work times and exception times. The work times are defined as recurring patterns with optional boundaries (applying from and/or to a specific date). The shown example defines a simple work calendar with working times Monday to Thursday 8:00 to 12:00 and 13:00 to 17:00, Friday 8:00 to 14:00 and as exception every 1st Monday in a month the work starts one hour later - i.e. the working time on every 1st Monday in a month is overridden to be 9:00 to 12:00 and 13:00 to 17:00. Both the working time and the exception time are valid for the period of 01.09.2010 till 30.08.2011.

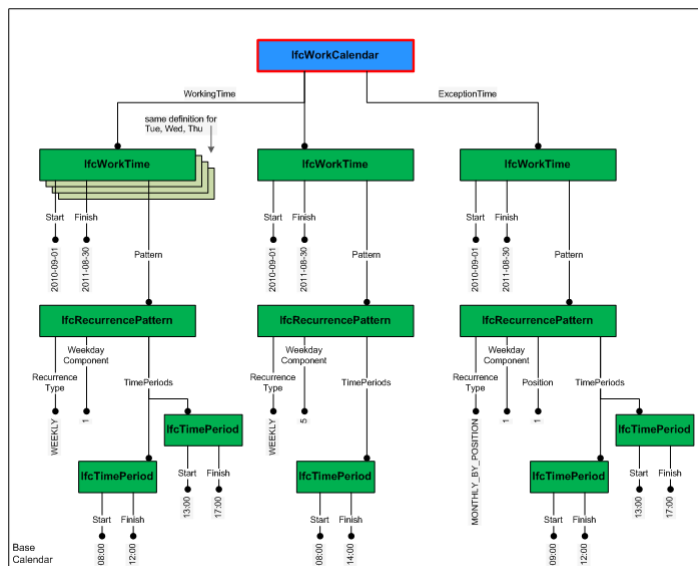


Figure 87 — Work calendar instantiation

EXPRESS Specification:

ENTITY IfcWorkCalendar

SUBTYPE OF (IfcControl);

WorkingTimes : **OPTIONAL SET** [1:?] **OF** IfcWorkTime;
 ExceptionTimes : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;
 PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcWorkCalendarTypeEnum.USERDEFINED)
 OR ((PredefinedType = IfcWorkCalendarTypeEnum.USERDEFINED) AND
 EXISTS(SELF\IfcObject.ObjectType));

END_ENTITY;Attribute Definitions:

WorkingTimes : Set of times periods that are regarded as an initial set-up of working times. Exception times can then further restrict these working times.
ExceptionTimes : Set of times periods that define exceptions (non-working times) for the given working times including the base calendar, if provided.

Formal Propositions:

CorrectPredefinedType : The attribute ObjectType must be asserted when the value of the IfcWorkCalendarTypeEnum is set to USERDEFINED.

Inheritance Graph:**ENTITY** IfcWorkCalendar**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] **OF** IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] **OF** IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] **OF** IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcControl

Identification : **OPTIONAL** IfcStrippedOptional;

INVERSE

Controls : **SET OF** IfcRelAssignsToControl **FOR** RelatingControl;

ENTITY IfcWorkCalendar

WorkingTimes : **OPTIONAL SET** [1:?] **OF** IfcWorkTime;

ExceptionTimes : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcWorkCalendar" type="ifc:IfcWorkCalendar" substitutionGroup="ifc:IfcControl" nillable="true"/>
<xs:complexType name="IfcWorkCalendar">
  <xs:complexContent>
    <xs:extension base="ifc:IfcControl">
      <xs:sequence>
        <xs:element name="WorkingTimes" nillable="true" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="ifc:IfcWorkTime" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute ref="ifc:itemType" fixed="ifc:IfcWorkTime"/>
            <xs:attribute ref="ifc:cType" fixed="set"/>
            <xs:attribute ref="ifc:arraySize" use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcWorkPlan

An *IfcWorkPlan* represents work plans in a construction or a facilities management project.

HISTORY New entity in IFC2.0

A work plan contains a set of work schedules for different purposes (including construction and facilities management). Contained work schedules are defined through the *IfcRelAggregates* relationship. Through inheritance from *IfcWorkControl* it is also possible to define references to activities (for example, *IfcTask*) and resources used in the work plan.

A work plan has information such as start date, finish date, total free float, and so on. *IfcWorkPlan* can also refer to the construction project represented by the single *IfcProject* instance (please also check the definition of *IfcWorkControl*).

Figure 88 shows the backbone structure of a work plan that defines (1) contained work schedules through *IfcRelAggregates* and (2), if not assigned otherwise to contained work schedules, assigned tasks and resources through *IfcRelAssignsToControl*.

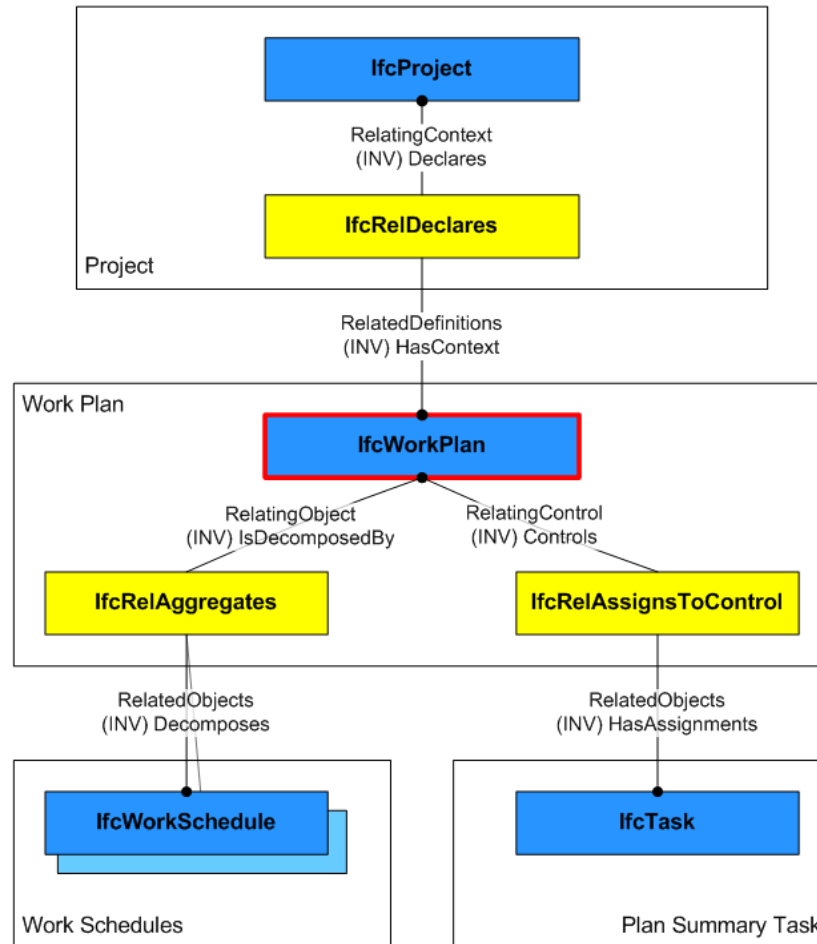


Figure 88 — Work plan relationships

[EXPRESS Specification:](#)

ENTITY *IfcWorkPlan*

SUBTYPE OF (*IfcWorkControl*);

PredefinedType : **OPTIONAL** *IfcStrippedOptional*;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> *IfcWorkPlanTypeEnum.USERDEFINED*) OR ((PredefinedType = *IfcWorkPlanTypeEnum.USERDEFINED*) AND EXISTS(SELF.*IfcObject.ObjectType*));

END_ENTITY;

Attribute Definitions:

Formal Propositions:

CorrectPredefinedType : The attribute ObjectType must be asserted when the value of the IfcWorkPlanTypeEnum is set to USERDEFINED.

Inheritance Graph:

ENTITY IfcWorkPlan

ENTITY IfcRoot

GlobalId	: IfcGloballyUniqueId;
OwnerHistory	: OPTIONAL IfcOwnerHistory;
Name	: OPTIONAL IfcLabel;
Description	: OPTIONAL IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments	: SET OF IfcRelAssigns FOR RelatedObjects;
Nests	: SET [0:1] OF IfcRelNests FOR RelatedObjects;
IsNestedBy	: SET OF IfcRelNests FOR RelatingObject;
HasContext	: SET [0:1] OF IfcRelDeclares FOR RelatedDefinitions;
IsDecomposedBy	: SET OF IfcRelAggregates FOR RelatingObject;
Decomposes	: SET [0:1] OF IfcRelAggregates FOR RelatedObjects;
HasAssociations	: SET OF IfcRelAssociates FOR RelatedObjects;

ENTITY IfcObject

ObjectType	: OPTIONAL IfcStrippedOptional;
------------	--

INVERSE

IsDeclaredBy	: SET [0:1] OF IfcRelDefinesByObject FOR RelatedObjects;
Declares	: SET OF IfcRelDefinesByObject FOR RelatingObject;
IsTypedBy	: SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;
IsDefinedBy	: SET OF IfcRelDefinesByProperties FOR RelatedObjects;

ENTITY IfcControl

Identification : **OPTIONAL** IfcStrippedOptional;

INVERSE

Controls : **SET OF** IfcRelAssignsToControl **FOR** RelatingControl;

ENTITY IfcWorkControl

CreationDate : IfcDateTime;
 Creators : **OPTIONAL SET** [1:?] **OF** IfcStrippedOptional;
 Purpose : **OPTIONAL** IfcStrippedOptional;
 Duration : **OPTIONAL** IfcStrippedOptional;
 TotalFloat : **OPTIONAL** IfcStrippedOptional;
 StartTime : IfcDateTime;
 FinishTime : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcWorkPlan

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcWorkPlan" type="ifc:IfcWorkPlan" substitutionGroup="ifc:IfcWorkControl" nillable="true"/>
<xs:complexType name="IfcWorkPlan">
  <xs:complexContent>
    <xs:extension base="ifc:IfcWorkControl"/>
  </xs:complexContent>
</xs:complexType>
```

IfcBuilding

A building represents a structure that provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, storey, and space).

NOTE Definition from ISO 6707-1:

Construction work that has the provision of shelter for its occupants or contents as one of its main purpose and is normally designed to stand permanently in one place.

A building is (if specified) associated to a site. A building may span over several connected or disconnected buildings. Therefore building complex provides for a collection of buildings included in a site. A building can also be decomposed in (vertical) parts, where each part defines a building section. This is defined by the composition type attribute of the supertype *IfcSpatialStructureElements* which is interpreted as follow:

- **COMPLEX:** building complex
- **ELEMENT:** building
- **PARTIAL:** building section

The *IfcBuilding* is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship *IfcRelAggregates*.

HISTORY New entity in IFC1.0.

Figure 27 shows the *IfcBuilding* as part of the spatial structure. It also serves as the spatial container for building and other elements.

NOTE Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

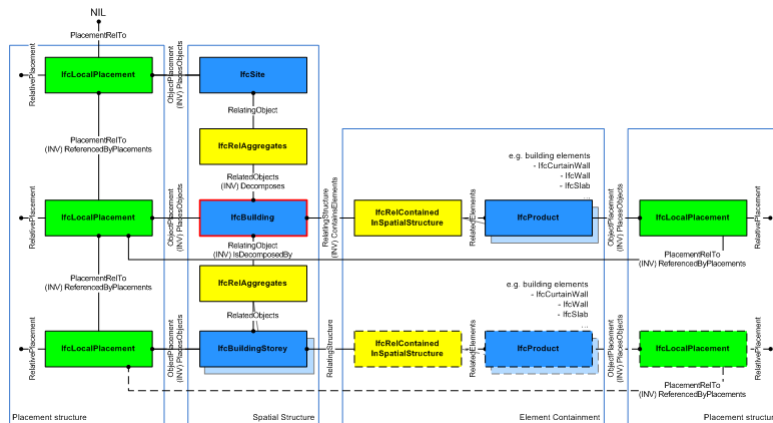


Figure 27 — Building composition

Systems, such as building service or electrical distribution systems, zonal systems, or structural analysis systems, relate to *IfcBuilding* by using the objectified relationship *IfcRelServicesBuildings*.

Attribute Use Definition

Figure 28 describes the heights and elevations of the *IfcBuilding*. It is used to provide the height above sea level of the project height datum for this building, that is, the internal height 0.00. The height 0.00 is often used as a building internal reference height and equal to the floor finish level of the ground floor.

- base elevation of building provided by: *IfcBuilding.ElevationOfRefHeight*, it is usually the top of construction slab
- base elevation of terrain at the perimeter of the building provided by: *IfcBuilding.ElevationOfTerrain*, it is usually the minimum elevation is sloped terrain
- total height of building, also referred to as ridge height (top of roof structure, e.g the ridge against terrain): provided by BaseQuantity with Name="TotalHeight"
- eaves height of building (base of roof structure, e.g the eaves against terrain): provided by BaseQuantity with Name="EavesHeight"

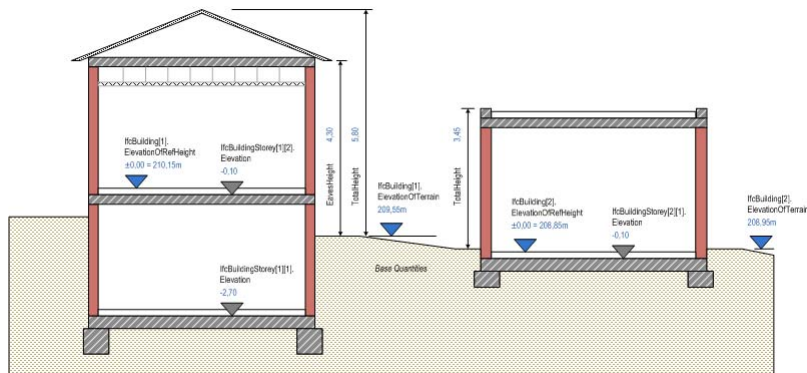


Figure 28 — Building elevations

EXPRESS Specification:

ENTITY IfcBuilding

SUBTYPE OF (IfcSpatialStructureElement);

ElevationOfRefHeight : **OPTIONAL** IfcLengthMeasure;
 ElevationOfTerrain : **OPTIONAL** IfcLengthMeasure;
 BuildingAddress : **OPTIONAL** IfcPostalAddress;

END_ENTITY;

Attribute Definitions:

ElevationOfRefHeight : Elevation above sea level of the reference height used for all storey elevation measures, equals to height 0.0. It is usually the ground floor level.
ElevationOfTerrain : Elevation above the minimal terrain level around the foot print of the building, given in elevation above sea level.
BuildingAddress : Address given to the building for postal purposes.

Inheritance Graph:

ENTITY IfcBuilding

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

INVERSE

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

ENTITY IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcBuilding

ElevationOfRefHeight : **OPTIONAL** IfcLengthMeasure;
 ElevationOfTerrain : **OPTIONAL** IfcLengthMeasure;
 BuildingAddress : **OPTIONAL** IfcPostalAddress;

END_ENTITY;

```
<xs:element name="IfcBuilding" type="ifc:IfcBuilding" substitutionGroup="ifc:IfcSpatialStructureElement" nillable="true"/>
<xs:complexType name="IfcBuilding">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement">
      <xs:sequence>
        <xs:element name="BuildingAddress" type="ifc:IfcPostalAddress" nillable="true" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="ElevationOfRefHeight" type="ifc:IfcLengthMeasure" use="optional"/>
      <xs:attribute name="ElevationOfTerrain" type="ifc:IfcLengthMeasure" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcBuildingStorey

The building storey has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound.

A storey is (if specified) associated to a building. A storey may span over several connected storeys. Therefore storey complex provides for a collection of storeys included in a building. A storey can also be decomposed in (horizontal) parts, where each part defines a partial storey. This is defined by the composition type attribute of the supertype *IfcSpatialStructureElements* which is interpreted as follow:

- **COMPLEX:** building storey complex

- **ELEMENT:** building storey

- **PARTIAL:** partial building storey

EXAMPLE In split level houses, a storey is split into two or more partial storeys, each with a different elevation. It can be handled by defining a storey, which includes two or more partial storeys with the individual elevations.

The *IfcBuildingStorey* is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship *IfcRelAggregates*.

Figure 29 shows the *IfcBuildingStorey* as part of the spatial structure. It also serves as the spatial container for building and other elements.

NOTE Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

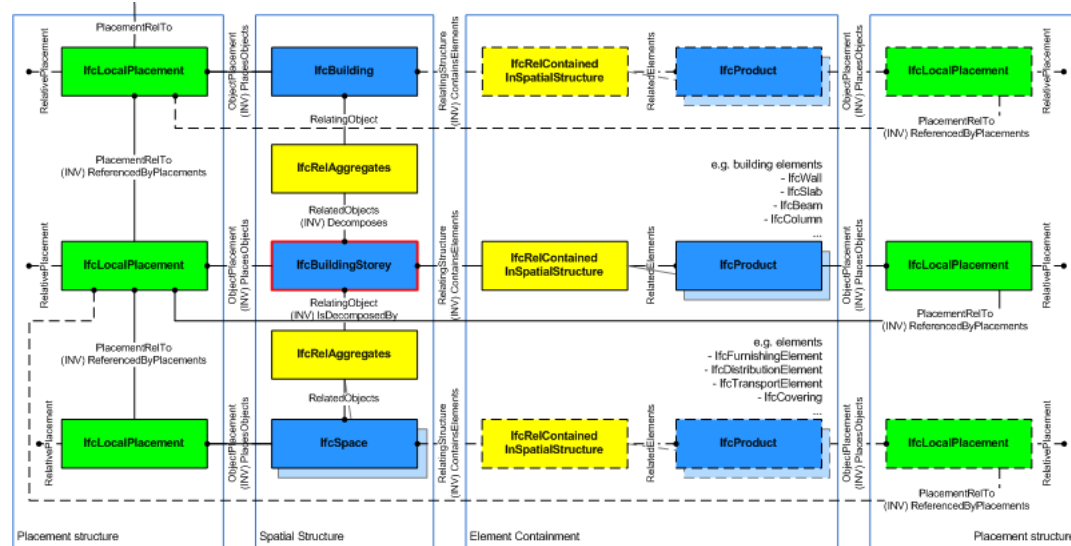


Figure 29 — Building storey composition

HISTORY New entity in IFC1.0

Attribute Use Definition

Figure 30 describes the heights and elevations of the *IfcBuildingStorey*.

- elevation of storey provided by: *IfcBuildingStorey.Elevation* as a local height value relative to *IfcBuilding.ElevationOfRefHeight*, it is usually the top of construction slab
- net height of storey, also referred to as total height or system height (top of construction slab to top of construction slab above): provided by BaseQuantity with Name="GrossHeight"
- net height of storey (top of construction slab to bottom of construction slab above): provided by BaseQuantity with Name="NetHeight"

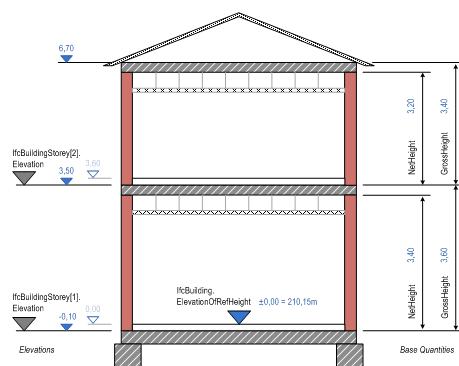


Figure 30 — Building storey elevations[EXPRESS Specification:](#)**ENTITY** IfcBuildingStorey**SUBTYPE OF** (IfcSpatialStructureElement);Elevation : **OPTIONAL** IfcStrippedOptional;**END_ENTITY;**[Attribute Definitions:](#)

Elevation : Elevation of the base of this storey, relative to the 0,00 internal reference height of the building. The 0.00 level is given by the absolute above sea level height by the ElevationOfRefHeight attribute given at IfcBuilding.

[Inheritance Graph:](#)**ENTITY** IfcBuildingStorey**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObjectObjectType : **OPTIONAL** IfcStrippedOptional;**INVERSE**

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

INVERSE

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

ENTITY IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcBuildingStorey

Elevation : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcBuildingStorey" type="ifc:IfcBuildingStorey" substitutionGroup="ifc:IfcSpatialStructureElement"
nillable="true"/>
<xs:complexType name="IfcBuildingStorey">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement"/>
  </xs:complexContent>
</xs:complexType>

```

IfcDistributionElement

This entity is a generalization of all elements that participate in a distribution system. Typical examples of *IfcDistributionElement* are (among others):

- building service elements within a heating systems
- building service elements within a cooling system
- building service elements within a ventilation system
- building service elements within a plumbing system

- electrical elements
- elements within a communication network

The *IfcDistributionElement* is further specialized in the IFC model. Direct instantiation of *IfcDistributionElement* without an assigned subtype of *IfcDistributionElementType* provides the meaning of an distribution element proxy.

HISTORY New entity in IFC1.5.

IFC4 CHANGE The entity is marked as deprecated for instantiation - will be made ABSTRACT in future releases.

EXPRESS Specification:

ENTITY IfcDistributionElement

SUPERTYPE OF (IfcDistributionFlowElement)

SUBTYPE OF (IfcElement);

INVERSE

END_ENTITY;

Attribute Definitions:

Inheritance Graph:

ENTITY IfcDistributionElement

ENTITY IfcRoot

GlobalId	: IfcGloballyUniqueId;
OwnerHistory	: OPTIONAL IfcOwnerHistory;
Name	: OPTIONAL IfcLabel;
Description	: OPTIONAL IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments	: SET OF IfcRelAssigns FOR RelatedObjects;
Nests	: SET [0:1] OF IfcRelNests FOR RelatedObjects;
IsNestedBy	: SET OF IfcRelNests FOR RelatingObject;
HasContext	: SET [0:1] OF IfcRelDeclares FOR RelatedDefinitions;
IsDecomposedBy	: SET OF IfcRelAggregates FOR RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****END_ENTITY;**

```
<xs:element name="IfcDistributionElement" type="ifc:IfcDistributionElement" substitutionGroup="ifc:IfcElement"
nillable="true"/>
<xs:complexType name="IfcDistributionElement">
  <xs:complexContent>
    <xs:extension base="ifc:IfcElement"/>
  </xs:complexContent>
</xs:complexType>
```

```
</xs:complexContent>
</xs:complexType>
```

IfcGrid

IfcGrid is a planar design grid defined in 3D space used as an aid in locating structural and design elements. The position of the grid (*ObjectPlacement*) is defined by a 3D coordinate system (and thereby the design grid can be used in plan, section or in any position relative to the world coordinate system). The position can be relative to the object placement of other products or grids. The XY plane of the 3D coordinate system is used to place the grid axes, which are 2D curves (for example, line, circle, arc, polyline).

The inherited attributes *Name* and *Description* can be used to define a descriptive name of the grid and to indicate the grid's purpose. A grid is defined by (normally) two, or (in case of a triangular grid) three lists of grid axes. The following figures show some examples.

A grid may support a rectangular layout as shown in Figure 82, a radial layout as shown in Figure 83, or a triangular layout as shown in Figure 84.

NOTE The *PredefinedType* denotes the type of grid that is represented by *IfcGrid*. The instantiation of *IfcGridAxis*'s has to agree to the *PredefinedType*, if provided.

NOTE The grid axes, defined within the design grid, are those elements to which project objects will be placed relatively using the *IfcGridPlacement*.

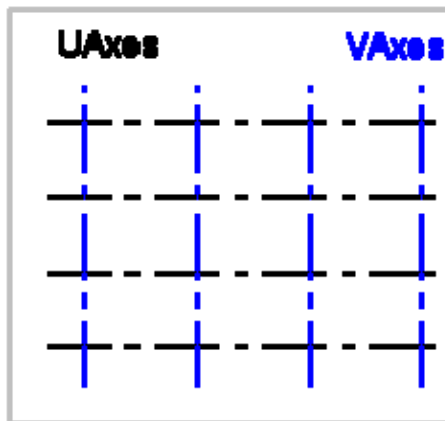


Figure 82 — Grid rectangular layout

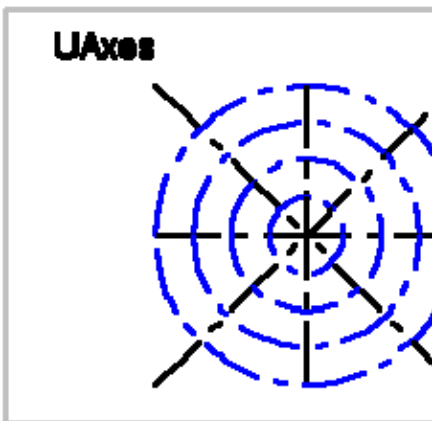


Figure 83 — Grid radial layout

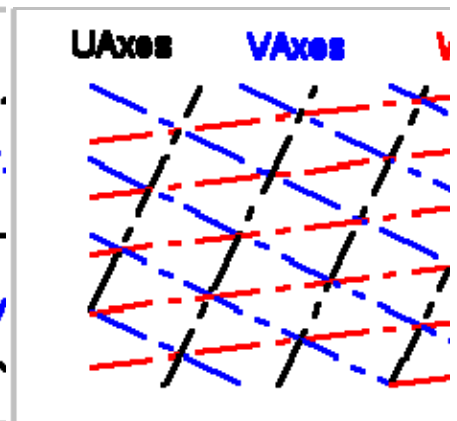


Figure 84 — Grid triangular layout

HISTORY New entity in IFC1.0.

IFC4 CHANGE The attribute *PredefinedType* has been added at the end of the attribute list.

Informal Propositions:

1. Grid axes, which are referenced in different lists of axes (UAxes, VAxes, WAxes) shall not be parallel.
2. Grid axes should be defined such as there are no two grid axes which intersect twice (see Figure 189).

NOTE Left side: ambiguous intersections A1 and A2, a grid containing such grid axes is not a valid design grid; Right side: the conflict can be resolved by splitting one grid axis in a way, such as no ambiguous intersections exist.

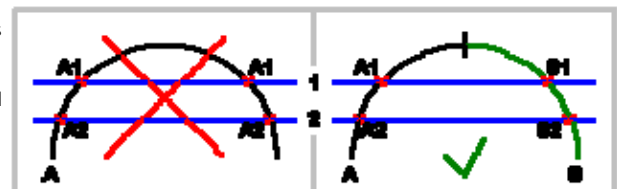


Figure 85 — Grid intersections

EXPRESS Specification:

ENTITY *IfcGrid*

SUBTYPE OF (*IfcProduct*);

UAxes : **LIST** [1:?] **OF UNIQUE** IfcGridAxis;
 VAxes : **LIST** [1:?] **OF UNIQUE** IfcGridAxis;
 WAxes : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 PredefinedType : **OPTIONAL** IfcGridTypeEnum;

INVERSE

ContainedInStructure : **SET** [0:1] **OF** IfcRelContainedInSpatialStructure **FOR** RelatedElements;

WHERE

HasPlacement : EXISTS(SELF\IfcProduct.ObjectPlacement);

END_ENTITY;

Attribute Definitions:

UAxes : List of grid axes defining the first row of grid lines.
VAxes : List of grid axes defining the second row of grid lines.
WAxes : List of grid axes defining the third row of grid lines. It may be given in the case of a triangular grid.
PredefinedType : Predefined types to define the particular type of the grid.
IFC4 CHANGE New attribute.

ContainedInStructure : Relationship to a spatial structure element, to which the grid is primarily associated.

Formal Propositions:

HasPlacement : The placement for the grid has to be given.

Inheritance Graph:

ENTITY IfcGrid

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] **OF** IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] **OF** IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcGrid

UAxes : **LIST** [1:?] **OF UNIQUE** IfcGridAxis;
 VAxes : **LIST** [1:?] **OF UNIQUE** IfcGridAxis;
 WAxes : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 PredefinedType : **OPTIONAL** IfcGridTypeEnum;

INVERSE

ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

END_ENTITY;

```
<xs:element name="IfcGrid" type="Ifc: IfcGrid" substitutionGroup="Ifc: IfcProduct" nillable="true"/>
<xs:complexType name="IfcGrid">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcProduct">
      <xs:sequence>
        <xs:element name="UAxes">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="Ifc: IfcGridAxis" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:attribute ref="Ifc: itemType" fixed="Ifc: IfcGridAxis"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

<xs:attribute ref="ifc:cType" fixed="list-unique"/>
<xs:attribute ref="ifc:arraySize" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="VAxes">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ifc:IfcGridAxis" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="ifc:itemType" fixed="ifc:IfcGridAxis"/>
    <xs:attribute ref="ifc:cType" fixed="list-unique"/>
    <xs:attribute ref="ifc:arraySize" use="optional"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="PredefinedType" type="ifc:IfcGridTypeEnum" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

IfcSite

A site is a defined area of land, possibly covered with water, on which the project construction is to be completed. A site may be used to erect, retrofit or turn down building(s), or for other construction related developments.

NOTE Term according to ISO6707-1 vocabulary "area of land or water where construction work or other development is undertaken".

A site may include a definition of the single geographic reference point for this site (global position using WGS84 with *Longitude*, *Latitude* and *Elevation*). The precision is provided up to millionth of a second and it provides an absolute placement in relation to the real world as used in exchange with geospatial information systems. If asserted, the *Longitude*, *Latitude* and *Elevation* establish the point in WGS84 where the point 0.,0.,0. of the *LocalPlacement* of *IfcSite* is situated.

The geometrical placement of the site, defined by the *IfcLocalPlacement*, shall be always relative to the spatial structure element, in which this site is included, or absolute, i.e. to the world coordinate system, as established by the geometric representation context of the project. The world coordinate system, established at the *IfcProject.RepresentationContexts*, may include a definition of the true north within the XY plane of the world coordinate system, if provided, it can be obtained at *IfcGeometricRepresentationContext.TrueNorth*.

A project may span over several connected or disconnected sites. Therefore site complex provides for a collection of sites included in a project. A site can also be decomposed in parts, where each part defines a site section. This is defined by the composition type attribute of the supertype *IfcSpatialStructureElements* which is interpreted as follow:

- COMPLEX = site complex
- ELEMENT = site
- PARTIAL = site section

The *IfcSite* is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical).

Figure 32 shows the *IfcSite* as part of the spatial structure. In addition to the logical spatial structure, also the placement hierarchy is shown. In this example the spatial structure hierarchy and the placement hierarchy are identical.

NOTE Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

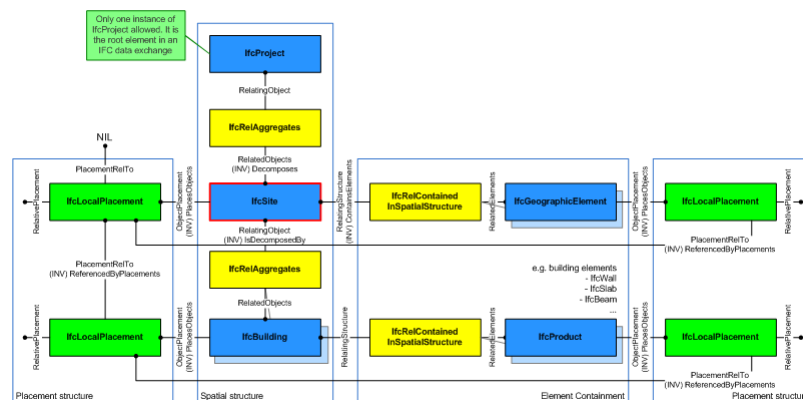


Figure 32 — Site composition

HISTORY New entity in IFC1.0.

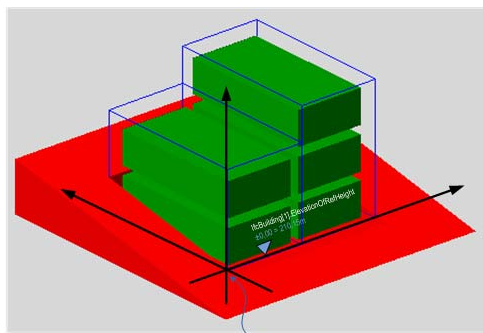
Attribute Use Definition

Figure 33 describes the heights and elevations of the *IfcSite*. It is used to provide the geographic longitude, latitude, and height above sea level for the origin of the site. The origin of the site is the local placement.

The provision of longitude, latitude, height at the *IfcSite* for georeferencing is provided for upward compatibility reasons. It requires a single instance of *IfcSite* and WGS84 as coordinate reference system.

For exact georeferencing (or referencing to any other geographic coordinate system other than WGS84) the entities *IfcCoordinateReferenceSystem* and *IfcMapConversion* have to be used to define an exact mapping of the project engineering coordinate system to the geographic (or map) coordinate system.

- reference height of site is provided by: *IfcSite.RefElevation*, it is given according to the height datum used at this location.
- the reference height of each building situated at the site is given against the same height datum used at this location.
- the elevations of each storey belonging to each building are given as local height relative to the reference height of the building.

**Figure 33 — Site elevations**

EXPRESS Specification:

ENTITY *IfcSite*

SUBTYPE OF (*IfcSpatialStructureElement*);

RefLatitude	: OPTIONAL <i>IfcCompoundPlaneAngleMeasure</i> ;
RefLongitude	: OPTIONAL <i>IfcCompoundPlaneAngleMeasure</i> ;
RefElevation	: OPTIONAL <i>IfcLengthMeasure</i> ;
LandTitleNumber	: OPTIONAL <i>IfcLabel</i> ;
SiteAddress	: OPTIONAL <i>IfcPostalAddress</i> ;

END_ENTITY;

Attribute Definitions:

RefLatitude	: World Latitude at reference point (most likely defined in legal description). Defined as integer values for degrees, minutes, seconds, and, optionally, millionths of seconds with respect to the world geodetic system WGS84. NOTE Latitudes are measured relative to the geodetic equator, north of the equator by positive values - from 0 till +90, south of the equator by negative values - from 0 till -90.
RefLongitude	: World Longitude at reference point (most likely defined in legal description). Defined as integer

values for degrees, minutes, seconds, and, optionally, millionths of seconds with respect to the world geodetic system WGS84.

NOTE Longitudes are measured relative to the geodetic zero meridian, nominally the same as the Greenwich prime meridian: longitudes west of the zero meridian have negative values - from 0 till -180, longitudes east of the zero meridian have positive values - from 0 till +180.

EXAMPLE Chicago Harbor Light has according to WGS84 a longitude -87.35.40 (or 87.35.40W) and a latitude 41.53.30 (or 41.53.30N).

RefElevation : Datum elevation relative to sea level.
LandTitleNumber : The land title number (designation of the site within a regional system).
SiteAddress : Address given to the site for postal purposes.

Inheritance Graph:

ENTITY IfcSite

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;

Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

INVERSE

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

ENTITY IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcSite

RefLatitude : **OPTIONAL** IfcCompoundPlaneAngleMeasure;

RefLongitude : **OPTIONAL** IfcCompoundPlaneAngleMeasure;

RefElevation : **OPTIONAL** IfcLengthMeasure;

LandTitleNumber : **OPTIONAL** IfcLabel;

SiteAddress : **OPTIONAL** IfcPostalAddress;

END_ENTITY;

```
<xs:element name="IfcSite" type="ifc: IfcSite" substitutionGroup="ifc: IfcSpatialStructureElement" nillable="true"/>
<xs:complexType name="IfcSite">
  <xs:complexContent>
    <xs:extension base="ifc: IfcSpatialStructureElement">
      <xs:sequence>
        <xs:element name="SiteAddress" type="ifc: IfcPostalAddress" nillable="true" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="RefLatitude" type="ifc: List- IfcCompoundPlaneAngleMeasure" use="optional"/>
      <xs:attribute name="RefLongitude" type="ifc: List- IfcCompoundPlaneAngleMeasure" use="optional"/>
      <xs:attribute name="RefElevation" type="ifc: IfcLengthMeasure" use="optional"/>
      <xs:attribute name="LandTitleNumber" type="ifc: IfcLabel" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcSpace

A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.

A space is associated to a building storey (or in case of exterior spaces to a site). A space may span over several connected spaces. Therefore a space group provides for a collection of spaces included in a storey. A space can also be decomposed in parts, where each part defines a partial space. This is defined by the *CompositionType* attribute of the supertype *IfcSpatialStructureElement* which is interpreted as follow:

- COMPLEX = space group

- ELEMENT = space

- PARTIAL = partial space

NOTE View definitions and implementation agreements may restrict spaces with *CompositionType*=ELEMENT to be non-overlapping.

The *IfcSpace* is used to build the spatial structure of a building (that serves as the primary project breakdown and is required to be hierarchical). The spatial structure elements are linked together by using the objectified relationship *IfcRelAggregates*.

Figure 34 shows the *IfcSpace* as part of the spatial structure. It also serves as the spatial container for space related elements.

NOTE Detailed requirements on mandatory element containment and placement structure relationships are given in view definitions and implementer agreements.

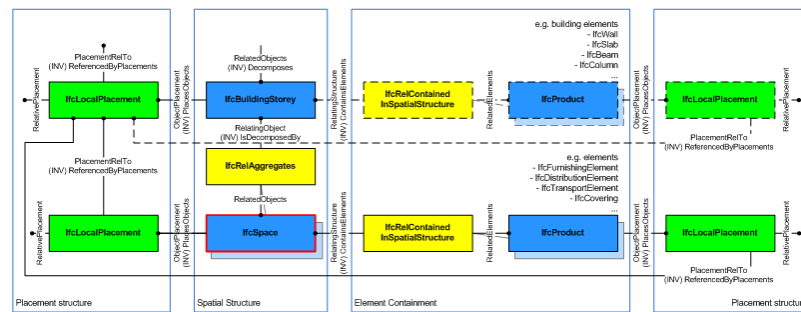


Figure 34 — Space composition

The following guidelines should apply for using the *Name*, *Description*, *LongName* and *ObjectType* attributes.

- *Name* holds the unique name (or space number) from the plan.
- *Description* holds any additional information field the user may have specified, there are no further recommendations.
- *LongName* holds the full name of the space, it is often used in addition to the *Name*, if a number is assigned to the room, then the descriptive name is exchanged as *LongName*.
- *ObjectType* holds the space type, i.e. usually the functional category of the space.

NOTE In cases of inconsistency between the geometric representation of the *IfcSpace* and the combined geometric representations of the surrounding *IfcRelSpaceBoundary*, the geometric representation of the space should take priority over the geometric representation of the surrounding space boundaries.

HISTORY New entity in IFC1.0

Attribute Use Definition

Figure 35 describes the heights and elevations of the *IfcSpace*.

- elevation of the space (top of construction slab) equals elevation of storey: provided by *IfcBuildingStorey.Elevation* relative to *IfcBuilding.ElevationOfRefHeight*
- elevation of the space flooring (top of flooring on top of slab): provided by *IfcSpace.ElevationWithFlooring* relative to *IfcBuilding.ElevationOfRefHeight*
- height of space (top of slab below to bottom of slab above): provided by BaseQuantity with Name="Height"
- floor height of space (top of slab below to top of flooring): provided by BaseQuantity with Name="FinishFloorHeight"
- net height of space (top of flooring to bottom of suspended ceiling): provided by BaseQuantity with Name="FinishCeilingHeight"

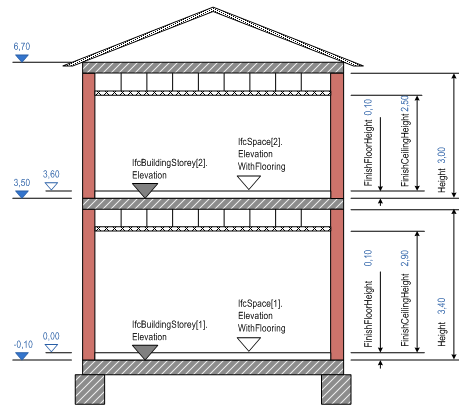


Figure 35 — Space elevations

EXPRESS Specification:

ENTITY IfcSpace

SUBTYPE OF (IfcSpatialStructureElement);

PredefinedType : **OPTIONAL** IfcStrippedOptional;
 ElevationWithFlooring : **OPTIONAL** IfcStrippedOptional;

INVERSE

HasCoverings : **SET OF** IfcRelCoversSpaces **FOR** RelatingSpace;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcSpaceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSpaceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
 CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCPRODUCTEXTENSION.IFCSPACETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Predefined generic types for a space that are specified in an enumeration. There might be property sets defined specifically for each predefined type.
 NOTE Previous use had been to indicate whether the IfcSpace is an interior space by value INTERNAL, or an exterior space by value EXTERNAL. This use is now deprecated, the property 'IsExternal' at 'Pset_SpaceCommon' should be used instead.

IFC4 CHANGE The attribute has been renamed from *ExteriorOrInteriorSpace* with upward compatibility for file based exchange.

ElevationWithFlooring : Level of flooring of this space; the average shall be taken, if the space ground surface is sloping or if there are level differences within this space.

HasCoverings : Reference to IfcCovering by virtue of the objectified relationship IfcRelCoversSpaces. It defines the concept of a space having coverings assigned. Those coverings may represent different flooring, or tiling areas.
 NOTE Coverings are often managed by the space, and not by the building element, which they

cover.

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcSpaceType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no space type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcSpaceType*.

Inheritance Graph:

ENTITY *IfcSpace*

ENTITY *IfcRoot*

GlobalId : *IfcGloballyUniqueId*;
 OwnerHistory : **OPTIONAL** *IfcOwnerHistory*;
 Name : **OPTIONAL** *IfcLabel*;
 Description : **OPTIONAL** *IfcText*;

ENTITY *IfcObjectDefinition*

INVERSE

HasAssignments : **SET OF** *IfcRelAssigns* **FOR** *RelatedObjects*;
 Nests : **SET** [0:1] OF *IfcRelNests* **FOR** *RelatedObjects*;
 IsNestedBy : **SET OF** *IfcRelNests* **FOR** *RelatingObject*;
 HasContext : **SET** [0:1] OF *IfcRelDeclares* **FOR** *RelatedDefinitions*;
 IsDecomposedBy : **SET OF** *IfcRelAggregates* **FOR** *RelatingObject*;
 Decomposes : **SET** [0:1] OF *IfcRelAggregates* **FOR** *RelatedObjects*;
 HasAssociations : **SET OF** *IfcRelAssociates* **FOR** *RelatedObjects*;

ENTITY *IfcObject*

ObjectType : **OPTIONAL** *IfcStrippedOptional*;

INVERSE

IsDeclaredBy : **SET** [0:1] OF *IfcRelDefinesByObject* **FOR** *RelatedObjects*;
 Declares : **SET OF** *IfcRelDefinesByObject* **FOR** *RelatingObject*;
 IsTypedBy : **SET** [0:1] OF *IfcRelDefinesByType* **FOR** *RelatedObjects*;
 IsDefinedBy : **SET OF** *IfcRelDefinesByProperties* **FOR** *RelatedObjects*;

ENTITY *IfcProduct*

ObjectPlacement : **OPTIONAL** *IfcObjectPlacement*;
 Representation : **OPTIONAL** *IfcProductRepresentation*;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcSpatialElement

LongName : **OPTIONAL** IfcStrippedOptional;

INVERSE

ContainsElements : **SET OF** IfcRelContainedInSpatialStructure **FOR** RelatingStructure;

ENTITY IfcSpatialStructureElement

CompositionType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcSpace

PredefinedType : **OPTIONAL** IfcStrippedOptional;

ElevationWithFlooring : **OPTIONAL** IfcStrippedOptional;

INVERSE

HasCoverings : **SET OF** IfcRelCoversSpaces **FOR** RelatingSpace;

END_ENTITY;

```
<xs:element name="IfcSpace" type="ifc:IfcSpace" substitutionGroup="ifc:IfcSpatialStructureElement" nillable="true"/>
<xs:complexType name="IfcSpace">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElement"/>
  </xs:complexContent>
</xs:complexType>
```

IfcSpaceType

A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.

The *IfcSpaceType* defines a list of commonly shared defines commonly shared information for occurrences of spaces. The set of shared information may include:

- common properties within shared property sets
- common shape representations

It is used to define an space specification (i.e. the specific space information, that is common to all occurrences of that space type. Space types may be exchanged without being already assigned to occurrences.

NOTE The space types are often used to represent space catalogues, less so for sharing a common representation map. Space types in a space catalogue share same space classification and a common set of space requirement properties.

The occurrences of *IfcSpaceType* are represented by instances of *IfcSpace*.

HISTORY New entity in IFC2x3.

IFC4 CHANGE The attribute *LongName* has been added to the end of the entity definition.

Geometry Use Definition

The *IfcSpaceType* may define the shared geometric representation for all space occurrences. The *RepresentationMaps* attribute refers to a list of *IfcRepresentationMap*'s, that allow for multiple geometric representations (e.g. with *IfcShapeRepresentation*'s having an *RepresentationIdentifier* 'Box', 'FootPrint', or 'Body').

NOTE The product representations are defined as representation maps (at the level of the supertype *IfcTypeProduct*, which gets assigned by an element occurrence instance through the *IfcShapeRepresentation.Item[1]* being an *IfcMappedItem*. However view definitions and implementer agreements may prevent the usage of shared geometry for spaces.

EXPRESS Specification:

ENTITY IfcSpaceType

SUBTYPE OF (IfcSpatialStructureElementType);

PredefinedType : IfcSpaceTypeEnum;
LongName : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcSpaceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSpaceTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcSpatialElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Predefined types to define the particular type of space. There may be property set definitions available for each predefined type.
LongName : Long name for a space type, used for informal purposes. It should be used, if available, in conjunction with the inherited *Name* attribute.
NOTE In many scenarios the *Name* attribute refers to the short name or number of a space type, and the *LongName* refers to the full descriptive name.

IFC4 CHANGE New attribute added at the end of entity definition.

Formal Propositions:

CorrectPredefinedType : The inherited attribute *ElementType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

Inheritance Graph:

ENTITY IfcSpaceType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcSpatialElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcSpatialStructureElementType**ENTITY** IfcSpaceType

PredefinedType : IfcSpaceTypeEnum;
 LongName : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcSpaceType" type="ifc:IfcSpaceType" substitutionGroup="ifc:IfcSpatialStructureElementType"
nillable="true"/>
<xs:complexType name="IfcSpaceType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSpatialStructureElementType">
      <xs:attribute name="PredefinedType" type="ifc:IfcSpaceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

</xs:extension>
</xs:complexContent>
</xs:complexType>

```

IfcTransportElement

A transport element is a generalization of all transport related objects that move people, animals or goods within a building or building complex. The *IfcTransportElement* defines the occurrence of a transport element, that (if given), is expressed by the *IfcTransportElementType*.

EXAMPLE Transportation elements include elevator (lift), escalator, moving walkway, etc.

NOTE More detailed equipment that may be a part of a transportation device, like a lifting hook, is defined as *IfcDiscreteAccessory*. It maybe included as a part of the *IfcTransportElement* by virtue of the objectified relationship *IfcRelAggregates*.

Depending on local classification systems transport elements and transportation systems in buildings are either considered as part of a building system, or as part of a building service system. Within IFC they are considered as part of a building system and may have to be mapped appropriately.

HISTORY New entity in IFC2x.

IFC2x CHANGE The attribute *PredefinedType* (previously *OperationType*) is made optional.

IFC4 CHANGE The last attributes *CapacityByWeight* and *CapacityByNumber* are removed, use *Pset_TransportElementCommon* instead.

EXPRESS Specification:

ENTITY IfcTransportElement

SUBTYPE OF (IfcElement);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcTransportElementTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTransportElementTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCPRODUCTEXTENSION.IFCTRANSPORTELEMENTTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Predefined generic types for a transportation element that are specified in an enumeration. There might be property sets defined specifically for each predefined type.

IFC4 CHANGE The attribute has been changed to be optional.

Formal Propositions:

CorrectPredefinedType : Either the *IfcTransportElement* attribute is unset (e.g. because an *IfcTransportElementType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to *USERDEFINED*.

CorrectTypeAssigned : Either there is no transport element type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcTransportElementType*.

Inheritance Graph:

ENTITY IfcTransportElement**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcTransportElement

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcTransportElement" type="ifc:IfcTransportElement" substitutionGroup="ifc:IfcElement" nillable="true"/>
<xs:complexType name="IfcTransportElement">
  <xs:complexContent>
    <xs:extension base="ifc:IfcElement"/>
  </xs:complexContent>
</xs:complexType>
```

IfcTransportElementType

The element type *IfcTransportElementType* defines commonly shared information for occurrences of transport elements. The set of shared information may include:

- common properties within shared property sets
- common material information
- common shape representations

It is used to define a transport element specification (i.e. the specific product information that is common to all occurrences of that beam type). Transport element types (or the instantiable subtypes) may be exchanged without being already assigned to occurrences.

The occurrences of the *IfcTransportElementType* are represented by instances of *IfcTransportElement* (or its subtypes).

HISTORY New entity in IFC2x2.

Geometry Use Definition

The *IfcTransportElementType* may define the shared geometric representation for all transport element occurrences. The *RepresentationMaps* attribute refers to a list of *IfcRepresentationMap*'s, that allow for multiple geometric representations (e.g. with *IfcShapeRepresentation*'s having an *RepresentationIdentifier* 'Box', 'FootPrint', or 'Body').

NOTE The product shape representations are defined as *RepresentationMaps* (attribute of the supertype *IfcTypeProduct*), which get assigned by an element occurrence instance through the *IfcShapeRepresentation.Item[n]* being an *IfcMappedItem*. See *IfcTypeProduct* for further information.

NOTE The values of attributes *RepresentationIdentifier* and *RepresentationType* of *IfcShapeRepresentation* are restricted in the same way as those for *IfcTransportElementType*.

EXPRESS Specification:**ENTITY** IfcTransportElementType**SUBTYPE OF** (IfcElementType);

PredefinedType : IfcTransportElementTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcTransportElementTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTransportElementTypeEnum.USERDEFINED) AND EXISTS(SELFIIfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Predefined types to define the particular type of the transport element. There may be property set definitions available for each predefined type.

Formal Propositions:

CorrectPredefinedType : The inherited attribute *ElementType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

Inheritance Graph:

ENTITY IfcTransportElementType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcTransportElementType

PredefinedType : IfcTransportElementTypeEnum;

END_ENTITY;

```
<xs:element name="IfcTransportElementType" type="ifc:IfcTransportElementType" substitutionGroup="ifc:IfcElementType"
nillable="true"/>
<xs:complexType name="IfcTransportElementType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcElementType">
      <xs:attribute name="PredefinedType" type="ifc:IfcTransportElementTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcCovering

A covering is an element which covers some part of another element and is fully dependent on that other element. The *IfcCovering* defines the occurrence of a covering type, that (if given) is expressed by the *IfcCoveringType*.

NOTE Definition according to ISO ISO 6707-1: final coverings and treatments of surfaces and their intersections.

Coverings are elements with relationships to the covered element and the space on the other side, they may contain openings, assigned by *IfcRelVoidsElement*, material information, assigned by *IfcRelAssociatesMaterial*, and others.

EXAMPLE Coverings include wall claddings, floorings, suspended ceilings, moldings and skirting boards.

NOTE A more basic information about claddings, floorings, and ceilings of a space can be attached to *IfcSpace*'s using the Pset_SpaceCommon properties. Then only a name can be provided and the covering quantities would be interpreted from the space quantities.

Coverings can be assigned to

- a space represented by *IfcSpace*
- using the inverse relationship *CoversSpaces* pointing to *IfcRelCoversSpaces*. The space is then accessible via *IfcRelCoversSpaces.RelatedSpace*. It defines to which space a covering is facing towards.
- a space boundary represented by *IfcRelSpaceBoundary*
- using the inverse relationship *ProvidesBoundaries* pointing to *IfcRelSpaceBoundary*. The space is then accessible via *IfcRelSpaceBoundary.RelatingSpace*.
- a building element represented by *IfcBuildingElement*
- using the inverse relationship *Covers* pointing to *IfcRelCoversBldgElements*. The building element is then accessible via *IfcRelCoversBldgElements.RelatingBuildingElement*.

NOTE The mere containment relationship between an *IfcCovering* and an *IfcSpace* is created by using *IfcRelContainedInSpatialStructure*

The following guideline shall apply:

- (default) if the space has coverings that may not have their own shape representation and no defined relationships to the building elements they cover, then the *IfcCovering* shall be assigned to *IfcSpace* using the *IfcRelCoversSpaces* relationship,
- if the space has coverings that have an own shape representation and the space has defined space boundaries, then the covering, which relates to that space, shall be contained in the space using *IfcRelContainedInSpatialStructure*. It may be assigned to the space boundaries using the *IfcRelSpaceBoundary*.
- if the covering does not relate to a space, then the covering should be assigned to the building element or a distribution element using the *IfcRelCoversBldgElements* relationship.

HISTORY New entity in IFC1.0.

IFC2x CHANGE The attribute *PredefinedType* is now optional and should only be inserted when no type information, given by *IfcCoveringType*, is assigned to the *IfcCovering* occurrence by *IfcRelDefinesByType*.

EXPRESS Specification:

ENTITY *IfcCovering*

SUBTYPE OF (*IfcBuildingElement*);

PredefinedType : **OPTIONAL** *IfcCoveringTypeEnum*;

INVERSE

CoversSpaces : **SET** [0:1] OF *IfcRelCoversSpaces* **FOR** *RelatedCoverings*;

WHERE

CorrectPredefinedType : NOT(EXISTS(*PredefinedType*)) OR (*PredefinedType* <> *IfcCoveringTypeEnum.USERDEFINED*) OR ((*PredefinedType* = *IfcCoveringTypeEnum.USERDEFINED*) AND EXISTS (*SELF**IfcObject.ObjectType*));

CorrectTypeAssigned : (SIZEOF(*IsTypedBy*) = 0) OR ('IFCSHAREDBLDGELEMENTS.IFCCOVERINGTYPE' IN TYPEOF(*SELF**IfcObject.IsTypedBy*[1].*RelatingType*));

END_ENTITY;

Attribute Definitions:

PredefinedType : Predefined types to define the particular type of the covering. There may be property set definitions available for each predefined type.
NOTE The *PredefinedType* shall only be used, if no *IfcCoveringType* is assigned, providing its own *IfcCoveringType.PredefinedType*.

CoversSpaces : Reference to the objectified relationship that handles the relationship of the covering to the covered space.

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcCoveringType* is associated), or the inherited attribute *ObjectType* shall be given, if the *PredefinedType* is set to *USERDEFINED*.

CorrectTypeAssigned : Either there is no covering type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcCoveringType*.

Inheritance Graph:

ENTITY IfcCovering**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcBuildingElement**INVERSE****ENTITY** IfcCovering

PredefinedType : **OPTIONAL** IfcCoveringTypeEnum;

INVERSE

CoversSpaces : **SET** [0:1] OF IfcRelCoversSpaces **FOR** RelatedCoverings;

END_ENTITY;

```
<xs:element name="IfcCovering" type="ifc:IfcCovering" substitutionGroup="ifc:IfcBuildingElement" nillable="true"/>
<xs:complexType name="IfcCovering">
  <xs:complexContent>
    <xs:extension base="ifc:IfcBuildingElement">
      <xs:attribute name="PredefinedType" type="ifc:IfcCoveringTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcSlabElementedCase

The *IfcSlabElementedCase* defines a slab with certain constraints for the provision of its components. The *IfcSlabElementedCase* handles all cases of slabs, that are decomposed into parts:

- having components being assigned to the *IfcSlabElementedCase* using the *IfcRelAggregates* relationship accessible by the inverse relationship *IsDecomposedBy*.
- applying the constraint that the parts within the decomposition shall be of type *IfcBeam*, *IfcMember*, *IfcPlate*, *IfcBuildingElementPart* or *IfcBuildingElementProxy*.

HISTORY New entity in IFC4.

Voiding Use Definition:

As shown in Figure 116, openings within the composite slab are directly assigned to *IfcSlabElementedCase* using *IfcRelVoidsElement* pointing to *IfcOpeningElement* and apply to all aggregated parts. If individual parts have cutting and other voiding features, then the decomposed parts have a separate voiding relationship *IfcRelVoidsElement* pointing to *IfcVoidingFeature*.

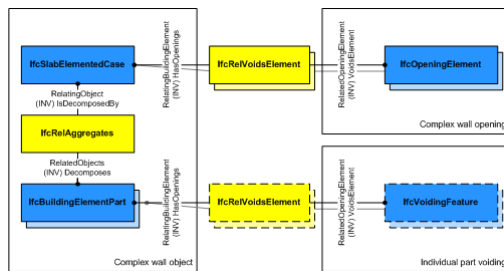


Figure 116 — Slab elemented voiding

EXPRESS Specification:

ENTITY IfcSlabElementedCase

SUBTYPE OF (IfcSlab);

WHERE

HasDecomposition : HIINDEX(SELf\IfcObjectDefinition.IsDecomposedBy) > 0;

END_ENTITY;

Formal Propositions:

HasDecomposition : A valid instance of IfcWallElementedCase has to have parts in a decomposition hierarchy.

Inheritance Graph:

ENTITY IfcSlabElementedCase

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] **OF** IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] **OF** IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcBuildingElement**INVERSE****ENTITY** IfcSlab

PredefinedType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcSlabElementedCase

END_ENTITY;

```

<xs:element name="IfcSlabElementedCase" type="ifc:IfcSlabElementedCase" substitutionGroup="ifc:IfcSlab" nillable="true"/>
<xs:complexType name="IfcSlabElementedCase">
  <xs:complexContent>
    <xs:extension base="ifc:IfcSlab"/>
  </xs:complexContent>
</xs:complexType>

```

IfcWallElementedCase

The *IfcWallElementedCase* defines a wall with certain constraints for the provision of its components. The *IfcWallElementedCase* handles all cases of walls, that are decomposed into parts:

- having components being assigned to the *IfcWallElementedCase* using the *IfcRelAggregates* relationship accessible by the inverse relationship *IsDecomposedBy*.
- applying the constraint that the parts within the decomposition shall be of type *IfcMember*, *IfcPlate*, *IfcBuildingElementPart* or *IfcBuildingElementProxy*.

HISTORY New entity in IFC4.

Property Set Use Definition:

The property sets relating to the *IfcWallElementedCase* are defined at the supertype *IfcWall*.

NOTE The parts within the decomposition relationship may define their own property sets.

Quantity Use Definition:

The quantities relating to the *IfcWallElementedCase* are defined at the supertype *IfcWall*.

NOTE The parts within the decomposition relationship may define their own individual quantities.

Voiding Use Definition:

As shown in Figure 121, openings within the composite wall are directly assigned to *IfcWallElementedCase* using *IfcRelVoidsElement* pointing to *IfcOpeningElement* and apply to all aggregated parts. If individual parts have cutting and other voiding features, then the decomposed parts have a separate voiding relationship *IfcRelVoidsElement* pointing to *IfcVoidingFeature*.

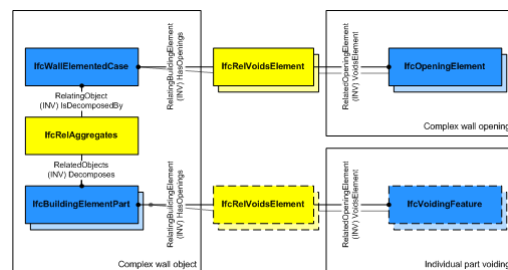


Figure 121 — Wall elemented voiding

Geometric Representation

The standard geometric representation of *IfcWallElementedCase* is defined using the following multiple shape representations for its definition:

- **Axis:** A two-dimensional open curve being a subtype of *IfcBoundedCurve* defining the axis for the elemented wall. It may be used as a simplified representation directly at the elemented wall.
- **Surface:** A three-dimensional surface being a subtype of *IfcBoundedSurface* defining the reference surface for the elemented wall. It may be used as a simplified representation directly at the elemented wall.

EXPRESS Specification:

ENTITY IfcWallElementedCase

SUBTYPE OF (IfcWall);

WHERE

HasDecomposition : HIINDEX(SELF\IfcObjectDefinition.IsDecomposedBy) > 0;

END_ENTITY;

Formal Propositions:

HasDecomposition : A valid instance of IfcWallElementedCase has to have parts in a decomposition hierarchy.

Inheritance Graph:

ENTITY IfcWallElementedCase

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcBuildingElement**INVERSE****ENTITY** IfcWall

PredefinedType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcWallElementedCase**END_ENTITY;**

```
<xs:element name="IfcWallElementedCase" type="ifc:IfcWallElementedCase" substitutionGroup="ifc:IfcWall" nillable="true"/>
<xs:complexType name="IfcWallElementedCase">
  <xs:complexContent>
    <xs:extension base="ifc:IfcWall"/>
  </xs:complexContent>
</xs:complexType>
```

IfcDistributionCircuit

A distribution circuit is a partition of a distribution system that is conditionally switched such as an electrical circuit.

HISTORY New entity in IFC4.

IFC4 CHANGE For electrical power systems, *IfcElectricalCircuit* has been used for low-voltage (12-1000 V) power circuits and has been replaced by *IfcDistributionCircuit* in IFC4; *IfcDistributionSystem* with PredefinedType 'ELECTRICAL' should be used for overall power

systems, and *IfcDistributionCircuit* with *PredefinedType* 'ELECTRICAL' should be used for each switched circuit.

Composition Use Definition

An *IfcDistributionCircuit* may aggregate an *IfcDistributionSystem* using the *IfcRelAggregates* relationship where *RelatingObject* refers to the *IfcDistributionSystem* and *RelatedObjects* includes one or more *IfcDistributionCircuit* groups.

An *IfcDistributionCircuit* may be aggregated into sub-circuits using the *IfcRelAggregates* relationship where *RelatingObject* refers to the parent *IfcDistributionCircuit* and *RelatedObjects* refers to one or more *IfcDistributionCircuit* sub-circuits.

Assignment Use Definition

An *IfcDistributionCircuit* should be assigned to an *IfcDistributionPort* on an *IfcFlowController* element indicating the host or origination of the system using the *IfcRelAssignsToProduct* relationship.

Each device whose operation is conditional based upon the state of the circuit is assigned to the *IfcDistributionCircuit* via the *IfcRelAssignsToGroup* relationship. An *IfcDistributionElement* may belong to multiple systems or circuits, however only one *IfcDistributionSystem* or *IfcDistributionCircuit* of a particular *PredefinedType*.

EXPRESS Specification:

ENTITY *IfcDistributionCircuit*

SUBTYPE OF (*IfcDistributionSystem*);

END_ENTITY;

Inheritance Graph:

ENTITY *IfcDistributionCircuit*

ENTITY *IfcRoot*

<i>GlobalId</i>	: <i>IfcGloballyUniqueId</i> ;
<i>OwnerHistory</i>	: OPTIONAL <i>IfcOwnerHistory</i> ;
<i>Name</i>	: OPTIONAL <i>IfcLabel</i> ;
<i>Description</i>	: OPTIONAL <i>IfcText</i> ;

ENTITY *IfcObjectDefinition*

INVERSE

<i>HasAssignments</i>	: SET OF <i>IfcRelAssigns</i> FOR <i>RelatedObjects</i> ;
<i>Nests</i>	: SET [0:1] OF <i>IfcRelNests</i> FOR <i>RelatedObjects</i> ;
<i>IsNestedBy</i>	: SET OF <i>IfcRelNests</i> FOR <i>RelatingObject</i> ;
<i>HasContext</i>	: SET [0:1] OF <i>IfcRelDeclares</i> FOR <i>RelatedDefinitions</i> ;
<i>IsDecomposedBy</i>	: SET OF <i>IfcRelAggregates</i> FOR <i>RelatingObject</i> ;
<i>Decomposes</i>	: SET [0:1] OF <i>IfcRelAggregates</i> FOR <i>RelatedObjects</i> ;
<i>HasAssociations</i>	: SET OF <i>IfcRelAssociates</i> FOR <i>RelatedObjects</i> ;

ENTITY *IfcObject*

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcGroup

INVERSE

IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

ENTITY IfcSystem

INVERSE

ENTITY IfcDistributionSystem

LongName : **OPTIONAL** IfcStrippedOptional;
 PredefinedType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionCircuit

END_ENTITY;

```
<xs:element name="IfcDistributionCircuit" type="ifc:IfcDistributionCircuit" substitutionGroup="ifc:IfcDistributionSystem"
nillable="true"/>
<xs:complexType name="IfcDistributionCircuit">
  <xs:complexContent>
    <xs:extension base="ifc:IfcDistributionSystem"/>
  </xs:complexContent>
</xs:complexType>
```

IfcDistributionPort

A distribution port is an inlet or outlet of a product through which a particular substance may flow.

Distribution ports are used for passage of solid, liquid, or gas substances, as well as electricity for power or communications. Flow segments (pipes, ducts, cables) may be used to connect ports across products. Distribution ports are defined by system type and flow direction such that for two ports to be connected, they must share the same system type and have opposite flow directions (one side being a *SOURCE* and the other being a *SINK*). Ports are similar to openings in that they do not have any visible geometry; such geometry is captured at the shape representation of the enclosing element or element type. Ports may have placement that indicates the position and orientation of the connection.

HISTORY [New entity in IFC2x2](#)

IFC4 CHANGE Ports are now related to products and product types using the *IfcRelNests* relationship; use of *IfcRelConnectsPortToElement* is now reserved for dynamically attached ports (such as drilling a hole in a tank).

EXPRESS Specification:

ENTITY IfcDistributionPort

SUBTYPE OF (IfcPort);

FlowDirection : **OPTIONAL** IfcFlowDirectionEnum;
 PredefinedType : **OPTIONAL** IfcDistributionPortTypeEnum;
 SystemType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

Attribute Definitions:

FlowDirection : Enumeration that identifies if this port is a Sink (inlet), a Source (outlet) or both a SinkAndSource.
SystemType : Enumeration that identifies the system type. If a system type is defined, the port may only be connected to other ports having the same system type.

Inheritance Graph:

ENTITY IfcDistributionPort

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcPort

INVERSE

ConnectedFrom : **SET** [0:1] OF IfcRelConnectsPorts **FOR** RelatedPort;
 ConnectedTo : **SET** [0:1] OF IfcRelConnectsPorts **FOR** RelatingPort;

ENTITY IfcDistributionPort

FlowDirection : **OPTIONAL** IfcFlowDirectionEnum;
 PredefinedType : **OPTIONAL** IfcDistributionPortTypeEnum;
 SystemType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcDistributionPort" type="ifc: IfcDistributionPort" substitutionGroup="ifc: IfcPort" nillable="true"/>
<xs:complexType name="IfcDistributionPort">
  <xs:complexContent>
    <xs:extension base="ifc: IfcPort">
      <xs:attribute name="FlowDirection" type="ifc: IfcFlowDirectionEnum" use="optional"/>
      <xs:attribute name="PredefinedType" type="ifc: IfcDistributionPortTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcDistributionSystem

A distribution system is a network designed to receive, store, maintain, distribute, or control the flow of a distribution media. A common example is a heating hot water system that consists of a pump, a tank, and an interconnected piping system for distributing hot water to terminals.

The group *IfcDistributionSystem* defines the occurrence of a specialized system for use within the context of building services.

Important functionalities for the description of a distribution system are derived from existing IFC entities:

- From *IfcSystem* it inherits the ability to couple the distribution system via *IfcRelServicesBuildings* to one or more *IfcSpatialElement* subtypes as necessary.
- From *IfcGroup* it inherits the inverse attribute *IsGroupedBy*, pointing to the relationship entity *IfcRelAssignsToGroup*. This allows to group distribution elements (instances of *IfcDistributionElement* subtypes), and in special cases ports directly (instances of *IfcDistributionPort*).
- From *IfcObject* it inherits the inverse attribute *IsDecomposedBy* pointing to the relationship entity *IfcRelAggregates*. It provides the hierarchy between the separate (partial) distribution systems. For example, an electrical main circuit may be aggregated into branch circuits.

HISTORY New entity in IFC4.

IFC4 CHANGE For electrical power systems, *IfcElectricalCircuit* has been used for low-voltage (12-1000 V) power circuits and has been replaced by *IfcDistributionCircuit* in IFC4; *IfcDistributionSystem* with PredefinedType 'ELECTRICAL' should be used for overall power systems, and *IfcDistributionCircuit* with PredefinedType 'ELECTRICAL' should be used for each switched circuit.

EXPRESS Specification:

ENTITY IfcDistributionSystem

SUPERTYPE OF (IfcDistributionCircuit)

SUBTYPE OF (IfcSystem);

LongName : **OPTIONAL** IfcStrippedOptional;
 PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

Attribute Definitions:

LongName : Long name for a system, used for informal purposes. It should be used, if available, in conjunction with the inherited *Name* attribute.
 NOTE In many scenarios the *Name* attribute refers to the short name or number of a distribution system or branch circuit, and the *LongName* refers to a descriptive name.

PredefinedType : Predefined types of distribution systems.

Inheritance Graph:

ENTITY IfcDistributionSystem

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcGroup**INVERSE**

IsGroupedBy : **SET OF** IfcRelAssignsToGroup **FOR** RelatingGroup;

ENTITY IfcSystem**INVERSE****ENTITY** IfcDistributionSystem

LongName : **OPTIONAL** IfcStrippedOptional;
 PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcDistributionSystem" type="ifc: IfcDistributionSystem" substitutionGroup="ifc: IfcSystem"
nillable="true"/>
<xs:complexType name="IfcDistributionSystem">
  <xs:complexContent>
    <xs:extension base="ifc: IfcSystem"/>
  </xs:complexContent>
</xs:complexType>

```

IfcOccupant

An occupant is a type of actor that defines the form of occupancy of a property.

The principal purpose of **IfcOccupant** is to determine the nature of occupancy of a property for a particular actor. All characteristics relating to the actor (name and organization details) are inherited from the *IfcActor* entity.

HISTORY New entity in IFC2x

EXPRESS Specification:

ENTITY IfcOccupant

SUBTYPE OF (IfcActor);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

WR31 : NOT(PredefinedType = IfcOccupantTypeEnum.USERDEFINED) OR
EXISTS(SELFIIfcObject.ObjectType);

END_ENTITY;

Attribute Definitions:

Formal Propositions:

WR31 : The attribute ObjectType must be asserted when the value of the IfcOccupantTypeEnum is set to USERDEFINED.

Inheritance Graph:

ENTITY IfcOccupant

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcActor

TheActor : IfcActorSelect;

INVERSE

IsActingUpon : **SET OF** IfcRelAssignsToActor **FOR** RelatingActor;

ENTITY IfcOccupant

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcOccupant" type="ifc:IfcOccupant" substitutionGroup="ifc:IfcActor" nillable="true"/>
<xs:complexType name="IfcOccupant">
  <xs:complexContent>
    <xs:extension base="ifc:IfcActor"/>
  </xs:complexContent>
</xs:complexType>

```

IfcCostItem

An *IfcCostItem* describes a cost or financial value together with descriptive information that describes its context in a form that enables it to be used within a cost schedule. An *IfcCostItem* can be used to represent the cost of goods and services, the execution of works by a process, lifecycle cost and more.

Each instance of *IfcCostItem* may have a name and a description. Depending on the use for which the cost is intended, these values should be asserted on the basis of agreement. For instance, the *Name* attribute could be used to provide a common value that enables distinct instances to be brought together in a nesting arrangement (see below) while the *Description* attribute may be used to provide text used for item description in a costing schedule.

An *IfcCostItem* can link one or many *IfcCostValue*'s representing a unit cost, total cost, or a unit cost with one or many quantities used to generate the total cost. The quantities can be given as individual quantities, or those quantities are provided as element quantities by one or many building elements. The *IfcCostValue.CostType* attribute indicates the category of cost, which may be used to present the value in a particular column. For nested cost items (having *IfcRelNests* relationship), *IfcCostValue.CostType* is significant such that *IfcCostValue.AppliedValue* is calculated as the sum of all nested costs having the same *IfcCostValue.CostType*.

or if set to an asterisk ('*'), then the sum of all nested costs of all cost types. An *IfcCostValue* may represent an original value or a value derived from formulas using *IfcAppliedValueRelationship*. For example, taxes may be calculated as a percentage of a subtotal.

HISTORY New entity in IFC2.0.

IFC4 CHANGE Attribute *PredefinedType*, *CostValues*, and *CostQuantities* added.

EXPRESS Specification:

ENTITY IfcCostItem

SUBTYPE OF (IfcControl);

PredefinedType : **OPTIONAL** IfcStrippedOptional;
 CostValues : **OPTIONAL LIST** [1:?] **OF** IfcCostValue;
 CostQuantities : **OPTIONAL LIST** [1:?] **OF** IfcStrippedOptional;

END_ENTITY;

Attribute Definitions:

CostValues : Component costs for which the total cost for the cost item is calculated, and then multiplied by the total *CostQuantities* if provided. If *CostQuantities* is provided then values indicate unit costs, otherwise values indicate total costs. For calculation purposes, the cost values may be directly added unless they have qualifications. Cost values with qualifications (e.g. *IfcCostValue.ApplicableDate*, *IfcCostValue.FixedUntilDate*) should be excluded from such calculation if they do not apply.
IFC4 CHANGE The attribute has been added.

CostQuantities : Component quantities of the same type for which the total quantity for the cost item is calculated as the sum.
IFC4 CHANGE The attribute has been added.

Inheritance Graph:

ENTITY IfcCostItem

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] **OF** IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] **OF** IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcControl

Identification : **OPTIONAL** IfcStrippedOptional;

INVERSE

Controls : **SET OF** IfcRelAssignsToControl **FOR** RelatingControl;

ENTITY IfcCostItem

PredefinedType : **OPTIONAL** IfcStrippedOptional;
 CostValues : **OPTIONAL LIST** [1:?] OF IfcCostValue;
 CostQuantities : **OPTIONAL LIST** [1:?] OF IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcCostItem" type="ifc:IfcCostItem" substitutionGroup="ifc:IfcControl" nillable="true"/>
<xs:complexType name="IfcCostItem">
  <xs:complexContent>
    <xs:extension base="ifc:IfcControl">
      <xs:sequence>
        <xs:element name="CostValues" nillable="true" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="ifc:IfcCostValue" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute ref="ifc:itemType" fixed="ifc:IfcCostValue"/>
            <xs:attribute ref="ifc:cType" fixed="list"/>
            <xs:attribute ref="ifc:arraySize" use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcCostSchedule

An *IfcCostSchedule* brings together instances of *IfcCostItem* either for the purpose of identifying purely cost information as in an estimate for constructions costs or for including cost information within another presentation form such as a work order.

HISTORY New entity in IFC2.0.

IFC4 CHANGE Attribute *ID* renamed to *Identification* and promoted to supertype *IfcControl*, *PredefinedType* made optional, attributes *PreparedBy*, *SubmittedBy*, *TargetUsers* removed.

EXPRESS Specification:

ENTITY IfcCostSchedule

SUBTYPE OF (IfcControl);

PredefinedType	: OPTIONAL IfcStrippedOptional;
Status	: OPTIONAL IfcStrippedOptional;
SubmittedOn	: OPTIONAL IfcStrippedOptional;
UpdateDate	: OPTIONAL IfcStrippedOptional;

END_ENTITY;

Attribute Definitions:

Status	: The current status of a cost schedule. Examples of status values that might be used for a cost schedule status include: PLANNED APPROVED AGREED ISSUED STARTED
SubmittedOn	: The date and time on which the cost schedule was submitted. IFC4 CHANGE Type changed from IfcDateTimeSelect.
UpdateDate	: The date and time that this cost schedule is updated; this allows tracking the schedule history. IFC4 CHANGE Type changed from IfcDateTimeSelect.

Inheritance Graph:

ENTITY IfcCostSchedule

ENTITY IfcRoot

GlobalId	: IfcGloballyUniqueId;
OwnerHistory	: OPTIONAL IfcOwnerHistory;
Name	: OPTIONAL IfcLabel;
Description	: OPTIONAL IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcControl

Identification : **OPTIONAL** IfcStrippedOptional;

INVERSE

Controls : **SET OF** IfcRelAssignsToControl **FOR** RelatingControl;

ENTITY IfcCostSchedule

PredefinedType : **OPTIONAL** IfcStrippedOptional;
 Status : **OPTIONAL** IfcStrippedOptional;
 SubmittedOn : **OPTIONAL** IfcStrippedOptional;
 UpdateDate : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcCostSchedule" type="Ifc: IfcCostSchedule" substitutionGroup="Ifc: IfcControl" nillable="true"/>
<xs:complexType name="IfcCostSchedule">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcControl"/>
  </xs:complexContent>
</xs:complexType>

```

IfcConstructionEquipmentResource

IfcConstructionEquipmentResource is usage of construction equipment to assist in the performance of construction. Construction Equipment resources are wholly or partially consumed or occupied in the performance of construction.

HISTORY New entity in IFC2.0.

Occurrences of *IfcConstructionEquipmentResource* are products that are used as resources to assist the process of construction. More specifically, they are products that are standalone items brought to a project to fulfil a particular purpose. Examples might be a tower crane or other mobile crane, a screwing machine, or a lifting hoist.

Instances of any subtype of *IfcProduct* may be assigned to the equipment resource using *IfcRelAssignsToResource* in order to characterize the equipment further, as described at the supertype *IfcResource*. Examples of relevant subtypes of *IfcProduct* are *IfcTransportElement*, *IfcDiscreteAccessory*, or *IfcProxy* (for particular cases where more precise usage details are not available)

EXPRESS Specification:

ENTITY IfcConstructionEquipmentResource

SUBTYPE OF (IfcConstructionResource);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcConstructionEquipmentResourceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcConstructionEquipmentResourceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines types of construction equipment resources.
IFC4 New attribute.

Inheritance Graph:

ENTITY IfcConstructionEquipmentResource

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0: 1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0: 1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0: 1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0: 1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcResource

Identification : **OPTIONAL** IfcStrippedOptional;
 LongDescription : **OPTIONAL** IfcStrippedOptional;

INVERSE

ResourceOf : **SET OF** IfcRelAssignsToResource **FOR** RelatingResource;

ENTITY IfcConstructionResource

Usage : **OPTIONAL** IfcStrippedOptional;
 BaseCosts : **OPTIONAL LIST** [1: ?] OF IfcAppliedValue;
 BaseQuantity : **OPTIONAL** IfcPhysicalQuantity;

ENTITY IfcConstructionEquipmentResource

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcConstructionEquipmentResource" type="ifc:IfcConstructionEquipmentResource"
substitutionGroup="ifc:IfcConstructionResource" nillable="true"/>
<xs:complexType name="IfcConstructionEquipmentResource">
  <xs:complexContent>
    <xs:extension base="ifc:IfcConstructionResource"/>
  </xs:complexContent>
</xs:complexType>

```

IfcSubContractResource

IfcSubContractResource is a construction resource needed in a construction process that represents a sub-contractor.

HISTORY New entity in IFC2.0.

IFC4 CHANGE The attribute *SubContractor* has been deleted; use *IfcRelAssignsToResource* to assign an *IfcActor* to fulfill the role as the subcontractor. The attribute *JobDescription* has been deleted; use *LongDescription* to describe the job.

An *IfcSubContractResource* can be used in cost estimating and work planning with or without specifying the subcontractor and contract agreement.

The purpose of an *IfcSubContractResource* is to indicate work of a particular type that is that is to be engaged through the use of a sub-contract. Its aim is to identify the description of the sub-contract work required. It can be used to identify the generic type of sub-contract resource that is required for a purpose without having to be specific about the actor (person or organization) providing the resource occurrence. It may be particularly useful when creating an overall plan for a process or processes. For instance, within maintenance or work planning there may be a known task that needs to be done which is planned to require an 'insulation specialist'.

A subcontract resource may be described at various stages and levels of detail through its assignments:

- Subcontract resource designated for particular tasks
- Actors identified to request bids
- Cost schedules (bids) received from actors
- Project order (work order, change order, etc.) executed

EXPRESS Specification:

ENTITY IfcSubContractResource

SUBTYPE OF (IfcConstructionResource);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcSubContractResourceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSubContractResourceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines types of subcontract resources.
IFC4 New attribute.

Inheritance Graph:

ENTITY IfcSubContractResource

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcResource

Identification : **OPTIONAL** IfcStrippedOptional;
 LongDescription : **OPTIONAL** IfcStrippedOptional;

INVERSE

ResourceOf : **SET OF** IfcRelAssignsToResource **FOR** RelatingResource;

ENTITY IfcConstructionResource

Usage : **OPTIONAL** IfcStrippedOptional;
 BaseCosts : **OPTIONAL LIST** [1:?] OF IfcAppliedValue;
 BaseQuantity : **OPTIONAL** IfcPhysicalQuantity;

ENTITY IfcSubContractResource

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcSubContractResource" type="ifc: IfcSubContractResource"
substitutionGroup="ifc: IfcConstructionResource" nillable="true"/>
<xs:complexType name="IfcSubContractResource">
  <xs:complexContent>
    <xs:extension base="ifc: IfcConstructionResource"/>
  </xs:complexContent>
</xs:complexType>

```

IfcSubContractResourceType

The resource type *IfcSubContractResourceType* defines commonly shared information for occurrences of subcontract resources. The set of shared information may include:

- common productivities
- common cost rates
- common properties within shared property sets

It is used to define a subcontract resource specification (the specific resource information that is common to all occurrences of that resource). Resource types may be exchanged without being already assigned to occurrences.

Occurrences of the *IfcSubContractResourceType* are represented by instances of *IfcSubContractResource*.

[HISTORY New entity in IFC4.](#)

[EXPRESS Specification:](#)

ENTITY IfcSubContractResourceType

SUBTYPE OF (IfcConstructionResourceType);

PredefinedType : IfcSubContractResourceTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcSubContractResourceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSubContractResourceTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcTypeResource.ResourceType));

END_ENTITY;

[Attribute Definitions:](#)

PredefinedType : Defines types of subcontract resources.

[Inheritance Graph:](#)

ENTITY IfcSubContractResourceType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0: 1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0: 1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0: 1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1: ?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0: 1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeResource

Identification : **OPTIONAL** IfcStrippedOptional;
 LongDescription : **OPTIONAL** IfcStrippedOptional;
 ResourceType : **OPTIONAL** IfcStrippedOptional;

INVERSE

ResourceOf : **SET OF** IfcRelAssignsToResource **FOR** RelatingResource;

ENTITY IfcConstructionResourceType

BaseCosts : **OPTIONAL LIST** [1: ?] OF IfcAppliedValue;
 BaseQuantity : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcSubContractResourceType

PredefinedType : IfcSubContractResourceTypeEnum;

END_ENTITY;

```

<xs:element name="IfcSubContractResourceType" type="ifc:IfcSubContractResourceType"
substitutionGroup="ifc:IfcConstructionResourceType" nillable="true"/>
<xs:complexType name="IfcSubContractResourceType">
  <xs:complexContent>

```

```

<xs:extension base="ifc:IfcConstructionResourceType">
  <xs:attribute name="PredefinedType" type="ifc:IfcSubContractResourceTypeEnum" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

IfcCableCarrierFitting

A cable carrier fitting is a fitting that is placed at junction or transition in a cable carrier system.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcCableCarrierFitting

SUBTYPE OF (IfcFlowFitting);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcCableCarrierFittingTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCableCarrierFittingTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCCABLECARRIERFITTINGTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of cable carrier fitting from which the type required may be set.

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcCableCarrierFittingType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no cable carrier fitting type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcCableCarrierFittingType.

Inheritance Graph:

ENTITY IfcCableCarrierFitting

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE**

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowFitting

ENTITY IfcCableCarrierFitting

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcCableCarrierFitting" type="ifc:IfcCableCarrierFitting" substitutionGroup="ifc:IfcFlowFitting"
nillable="true"/>
<xs:complexType name="IfcCableCarrierFitting">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowFitting"/>
  </xs:complexContent>
</xs:complexType>
```

IfcCableCarrierFittingType

The flow fitting type **IfcCableCarrierFittingType** defines commonly shared information for occurrences of cable carrier fittings. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a cable carrier fitting type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcCableCarrierFittingType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcCableCarrierFittingType** are represented by instances of *IfcCableCarrierFitting*. Refer to the documentation at *IfcCableCarrierFitting* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcCableCarrierFittingType

SUBTYPE OF (IfcFlowFittingType);

PredefinedType : IfcCableCarrierFittingTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcCableCarrierFittingTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCableCarrierFittingTypeEnum.USERDEFINED) AND EXISTS(SELf\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of cable carrier fitting from which the type required may be set.

Inheritance Graph:

ENTITY IfcCableCarrierFittingType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowFittingType**ENTITY** IfcCableCarrierFittingType

PredefinedType : IfcCableCarrierFittingTypeEnum;

END_ENTITY;

```
<xs:element name="IfcCableCarrierFittingType" type="ifc:IfcCableCarrierFittingType" substitutionGroup="ifc:IfcFlowFittingType"
nillable="true"/>
<xs:complexType name="IfcCableCarrierFittingType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowFittingType">
      <xs:attribute name="PredefinedType" type="ifc:IfcCableCarrierFittingTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcCableCarrierSegment

A cable carrier segment is a flow segment that is specifically used to carry and support cabling.

HISTORY New entity in IFC4

[EXPRESS Specification:](#)**ENTITY** IfcCableCarrierSegment**SUBTYPE OF** (IfcFlowSegment);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcCableCarrierSegmentTypeEnum.USERDEFINED) OR ((PredefinedType =
IfcCableCarrierSegmentTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICALDOMAIN.IFCCABLECARRIERSEGMENTTYPE' IN
TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;Attribute Definitions:

PredefinedType : Identifies the predefined types of cable carrier segment from which the type required may be set.

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcCableCarrierSegmentType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no cable carrier segment type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcCableCarrierSegmentType*.

Inheritance Graph:

ENTITY *IfcCableCarrierSegment*

ENTITY *IfcRoot*

GlobalId : *IfcGloballyUniqueId*;
 OwnerHistory : **OPTIONAL** *IfcOwnerHistory*;
 Name : **OPTIONAL** *IfcLabel*;
 Description : **OPTIONAL** *IfcText*;

ENTITY *IfcObjectDefinition*

INVERSE

HasAssignments : **SET OF** *IfcRelAssigns* **FOR** *RelatedObjects*;
 Nests : **SET** [0:1] OF *IfcRelNests* **FOR** *RelatedObjects*;
 IsNestedBy : **SET OF** *IfcRelNests* **FOR** *RelatingObject*;
 HasContext : **SET** [0:1] OF *IfcRelDeclares* **FOR** *RelatedDefinitions*;
 IsDecomposedBy : **SET OF** *IfcRelAggregates* **FOR** *RelatingObject*;
 Decomposes : **SET** [0:1] OF *IfcRelAggregates* **FOR** *RelatedObjects*;
 HasAssociations : **SET OF** *IfcRelAssociates* **FOR** *RelatedObjects*;

ENTITY *IfcObject*

ObjectType : **OPTIONAL** *IfcStrippedOptional*;

INVERSE

IsDeclaredBy : **SET** [0:1] OF *IfcRelDefinesByObject* **FOR** *RelatedObjects*;
 Declares : **SET OF** *IfcRelDefinesByObject* **FOR** *RelatingObject*;
 IsTypedBy : **SET** [0:1] OF *IfcRelDefinesByType* **FOR** *RelatedObjects*;
 IsDefinedBy : **SET OF** *IfcRelDefinesByProperties* **FOR** *RelatedObjects*;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowSegment**ENTITY** IfcCableCarrierSegment

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcCableCarrierSegment" type="ifc:IfcCableCarrierSegment" substitutionGroup="ifc:IfcFlowSegment"
  nillable="true"/>
<xs:complexType name="IfcCableCarrierSegment">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowSegment"/>
  </xs:complexContent>
</xs:complexType>
```

IfcCableCarrierSegmentType

The flow segment type **IfcCableCarrierSegmentType** defines commonly shared information for occurrences of cable carrier segments. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a cable carrier segment type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcCableCarrierSegmentType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcCableCarrierSegmentType** are represented by instances of *IfcCableCarrierSegment*. Refer to the documentation at *IfcCableCarrierSegment* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcCableCarrierSegmentType

SUBTYPE OF (IfcFlowSegmentType);

PredefinedType : IfcCableCarrierSegmentTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcCableCarrierSegmentTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCableCarrierSegmentTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

[Attribute Definitions:](#)

PredefinedType : Identifies the predefined types of cable carrier segment from which the type required may be set.

[Inheritance Graph:](#)

ENTITY IfcCableCarrierSegmentType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] OF **UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowSegmentType**ENTITY** IfcCableCarrierSegmentType

PredefinedType : IfcCableCarrierSegmentTypeEnum;

END_ENTITY;

```

<xs:element name="IfcCableCarrierSegmentType" type="ifc:IfcCableCarrierSegmentType"
substitutionGroup="ifc:IfcFlowSegmentType" nillable="true"/>
<xs:complexType name="IfcCableCarrierSegmentType">
  <xs:complexContent>

```

```

<xs:extension base="ifc:IfcFlowSegmentType">
  <xs:attribute name="PredefinedType" type="ifc:IfcCableCarrierSegmentTypeEnum" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

IfcCableFitting

A cable fitting is a fitting that is placed at a junction, transition or termination in a cable system.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcCableFitting

SUBTYPE OF (IfcFlowFitting);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcCableFittingTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCableFittingTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCCABLEFITTINGTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of cable fitting from which the type required may be set.

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcCableFittingType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no cable fitting type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcCableFittingType.

Inheritance Graph:

ENTITY IfcCableFitting

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0: 1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0: 1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0: 1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0: 1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0: 1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0: 1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE**

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowFitting

ENTITY IfcCableFitting

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcCableFitting" type="Ifc:IfcCableFitting" substitutionGroup="Ifc:IfcFlowFitting" nillable="true"/>
<xs:complexType name="IfcCableFitting">
  <xs:complexContent>
    <xs:extension base="Ifc:IfcFlowFitting"/>
  </xs:complexContent>
</xs:complexType>
```

IfcCableSegment

A cable segment is a flow segment used to carry electrical power, data, or telecommunications signals.

A cable segment is used to typically join two sections of an electrical network or a network of components carrying the electrical service.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcCableSegment

SUBTYPE OF (IfcFlowSegment);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcCableSegmentTypeEnum.USERDEFINED)
OR ((PredefinedType = IfcCableSegmentTypeEnum.USERDEFINED) AND EXISTS
(SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCCABLESEGMENTTYPE' IN
TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of cable segment from which the type required may be set.

Formal Propositions:

- CorrectPredefinedType** : Either the *PredefinedType* attribute is unset (e.g. because an *IfcCableSegmentType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.
- CorrectTypeAssigned** : Either there is no cable segment type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcCableSegmentType*.

Inheritance Graph:

ENTITY *IfcCableSegment*

ENTITY *IfcRoot*

GlobalId : *IfcGloballyUniqueId*;
 OwnerHistory : **OPTIONAL** *IfcOwnerHistory*;
 Name : **OPTIONAL** *IfcLabel*;
 Description : **OPTIONAL** *IfcText*;

ENTITY *IfcObjectDefinition*

INVERSE

HasAssignments : **SET OF** *IfcRelAssigns* **FOR** *RelatedObjects*;
 Nests : **SET** [0:1] OF *IfcRelNests* **FOR** *RelatedObjects*;
 IsNestedBy : **SET OF** *IfcRelNests* **FOR** *RelatingObject*;
 HasContext : **SET** [0:1] OF *IfcRelDeclares* **FOR** *RelatedDefinitions*;
 IsDecomposedBy : **SET OF** *IfcRelAggregates* **FOR** *RelatingObject*;
 Decomposes : **SET** [0:1] OF *IfcRelAggregates* **FOR** *RelatedObjects*;
 HasAssociations : **SET OF** *IfcRelAssociates* **FOR** *RelatedObjects*;

ENTITY *IfcObject*

ObjectType : **OPTIONAL** *IfcStrippedOptional*;

INVERSE

IsDeclaredBy : **SET** [0:1] OF *IfcRelDefinesByObject* **FOR** *RelatedObjects*;
 Declares : **SET OF** *IfcRelDefinesByObject* **FOR** *RelatingObject*;
 IsTypedBy : **SET** [0:1] OF *IfcRelDefinesByType* **FOR** *RelatedObjects*;
 IsDefinedBy : **SET OF** *IfcRelDefinesByProperties* **FOR** *RelatedObjects*;

ENTITY *IfcProduct*

ObjectPlacement : **OPTIONAL** *IfcObjectPlacement*;
 Representation : **OPTIONAL** *IfcProductRepresentation*;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;

InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;

ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowSegment

ENTITY IfcCableSegment

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcCableSegment" type="ifc:IfcCableSegment" substitutionGroup="ifc:IfcFlowSegment" nillable="true"/>
<xs:complexType name="IfcCableSegment">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowSegment"/>
  </xs:complexContent>
</xs:complexType>
```

IfcCableSegmentType

The flow segment type **IfcCableSegmentType** defines commonly shared information for occurrences of cable segments. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements

- common ports

It is used to define a cable segment type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcCableSegmentType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcCableSegmentType** are represented by instances of *IfcCableSegment*. Refer to the documentation at *IfcCableSegment* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcCableSegmentType

SUBTYPE OF (IfcFlowSegmentType);

PredefinedType : IfcCableSegmentTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcCableSegmentTypeEnum.USERDEFINED) OR ((PredefinedType = IfcCableSegmentTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of cable segment from which the type required may be set.

Inheritance Graph:

ENTITY IfcCableSegmentType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] **OF** IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowSegmentType**ENTITY** IfcCableSegmentType

PredefinedType : IfcCableSegmentTypeEnum;

END_ENTITY;

```
<xs:element name="IfcCableSegmentType" type="ifc:IfcCableSegmentType" substitutionGroup="ifc:IfcFlowSegmentType"
nillable="true"/>
<xs:complexType name="IfcCableSegmentType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowSegmentType">
      <xs:attribute name="PredefinedType" type="ifc:IfcCableSegmentTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricAppliance

An electric appliance is a device intended for consumer usage that is powered by electricity.

Electric appliances may be fixed in place or may be able to be moved from one space to another. Electric appliances require an electrical supply that may be supplied either by an electrical circuit or provided from a local battery source.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcElectricAppliance

SUBTYPE OF (IfcFlowTerminal);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcElectricApplianceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricApplianceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCELECTRICAPPLIANCETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcElectricApplianceType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no electric appliance type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcElectricApplianceType*.

Inheritance Graph:

ENTITY IfcElectricAppliance

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE**

ENTITY IfcFlowTerminal

ENTITY IfcElectricAppliance

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcElectricAppliance" type="ifc:IfcElectricAppliance" substitutionGroup="ifc:IfcFlowTerminal"
nillable="true"/>
<xs:complexType name="IfcElectricAppliance">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminal"/>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricApplianceType

The flow terminal type **IfcElectricApplianceType** defines commonly shared information for occurrences of electric appliances. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a electric appliance type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcElectricApplianceType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcElectricApplianceType** are represented by instances of *IfcElectricAppliance*. Refer to the documentation at *IfcElectricAppliance* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcElectricApplianceType

SUBTYPE OF (IfcFlowTerminalType);

PredefinedType : IfcElectricApplianceTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcElectricApplianceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricApplianceTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

[Attribute Definitions:](#)

PredefinedType : Identifies the predefined types of electrical appliance from which the type required may be set.

Inheritance Graph:**ENTITY** IfcElectricApplianceType**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcFlowTerminalType

ENTITY IfcElectricApplianceType

PredefinedType : IfcElectricApplianceTypeEnum;

END_ENTITY;

```
<xs:element name="IfcElectricApplianceType" type="ifc:IfcElectricApplianceType" substitutionGroup="ifc:IfcFlowTerminalType"
nillable="true"/>
<xs:complexType name="IfcElectricApplianceType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminalType">
      <xs:attribute name="PredefinedType" type="ifc:IfcElectricApplianceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricDistributionBoard

A distribution board is a flow controller in which instances of electrical devices are brought together at a single place for a particular purpose.

A distribution provides a housing for connected electrical distribution elements so that they can be viewed, operated or acted upon from a single place. Each connected item may have its own geometric representation and location.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcElectricDistributionBoard

SUBTYPE OF (IfcFlowController);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcElectricDistributionBoardTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricDistributionBoardTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCELECTRICDISTRIBUTIONBOARDTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcElectricDistributionBoardType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no electric distribution board type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcElectricDistributionBoardType*.

Inheritance Graph:

ENTITY *IfcElectricDistributionBoard*

ENTITY *IfcRoot*

GlobalId : *IfcGloballyUniqueId*;
 OwnerHistory : **OPTIONAL** *IfcOwnerHistory*;
 Name : **OPTIONAL** *IfcLabel*;
 Description : **OPTIONAL** *IfcText*;

ENTITY *IfcObjectDefinition*

INVERSE

HasAssignments : **SET OF** *IfcRelAssigns* **FOR** *RelatedObjects*;
 Nests : **SET** [0:1] OF *IfcRelNests* **FOR** *RelatedObjects*;
 IsNestedBy : **SET OF** *IfcRelNests* **FOR** *RelatingObject*;
 HasContext : **SET** [0:1] OF *IfcRelDeclares* **FOR** *RelatedDefinitions*;
 IsDecomposedBy : **SET OF** *IfcRelAggregates* **FOR** *RelatingObject*;
 Decomposes : **SET** [0:1] OF *IfcRelAggregates* **FOR** *RelatedObjects*;
 HasAssociations : **SET OF** *IfcRelAssociates* **FOR** *RelatedObjects*;

ENTITY *IfcObject*

ObjectType : **OPTIONAL** *IfcStrippedOptional*;

INVERSE

IsDeclaredBy : **SET** [0:1] OF *IfcRelDefinesByObject* **FOR** *RelatedObjects*;
 Declares : **SET OF** *IfcRelDefinesByObject* **FOR** *RelatingObject*;
 IsTypedBy : **SET** [0:1] OF *IfcRelDefinesByType* **FOR** *RelatedObjects*;
 IsDefinedBy : **SET OF** *IfcRelDefinesByProperties* **FOR** *RelatedObjects*;

ENTITY *IfcProduct*

ObjectPlacement : **OPTIONAL** *IfcObjectPlacement*;
 Representation : **OPTIONAL** *IfcProductRepresentation*;

INVERSE

ReferencedBy : **SET OF** *IfcRelAssignsToProduct* **FOR** *RelatingProduct*;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowController**ENTITY** IfcElectricDistributionBoard

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcElectricDistributionBoard" type="Ifc:IfcElectricDistributionBoard"
substitutionGroup="Ifc:IfcFlowController" nillable="true"/>
<xs:complexType name="IfcElectricDistributionBoard">
  <xs:complexContent>
    <xs:extension base="Ifc:IfcFlowController"/>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricDistributionBoardType

The flow controller type **IfcElectricDistributionBoardType** defines commonly shared information for occurrences of electric distribution boards. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define an electric distribution board type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcElectricDistributionBoardType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcElectricDistributionBoardType** are represented by instances of *IfcElectricDistributionBoard*. Refer to the documentation at *IfcElectricDistributionBoard* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcElectricDistributionBoardType

SUBTYPE OF (IfcFlowControllerType);

PredefinedType : IfcElectricDistributionBoardTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcElectricDistributionBoardTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricDistributionBoardTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of electric distribution type from which the type required may be set.

Inheritance Graph:

ENTITY IfcElectricDistributionBoardType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] **OF** IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowControllerType**ENTITY** IfcElectricDistributionBoardType

PredefinedType : IfcElectricDistributionBoardTypeEnum;

END_ENTITY;

```
<xs:element name="IfcElectricDistributionBoardType" type="ifc:IfcElectricDistributionBoardType"
substitutionGroup="ifc:IfcFlowControllerType" nillable="true"/>
<xs:complexType name="IfcElectricDistributionBoardType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowControllerType">
      <xs:attribute name="PredefinedType" type="ifc:IfcElectricDistributionBoardTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricFlowStorageDevice

An electric flow storage device is a device in which electrical energy is stored and from which energy may be progressively released.

HISTORY [New entity in IFC4](#)

[EXPRESS Specification:](#)

ENTITY IfcElectricFlowStorageDevice

SUBTYPE OF (IfcFlowStorageDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCELECTRICFLOWSTORAGEDEVICETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

[Formal Propositions:](#)

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcElectricFlowStorageDeviceType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no electric flow storage device type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcElectricFlowStorageDeviceType.

[Inheritance Graph:](#)

ENTITY IfcElectricFlowStorageDevice

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowStorageDevice

ENTITY IfcElectricFlowStorageDevice

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcElectricFlowStorageDevice" type="ifc:IfcElectricFlowStorageDevice"
substitutionGroup="ifc:IfcFlowStorageDevice" nillable="true"/>
<xs:complexType name="IfcElectricFlowStorageDevice">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowStorageDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricFlowStorageDeviceType

The flow storage device type **IfcElectricFlowStorageDeviceType** defines commonly shared information for occurrences of electric flow storage devices. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a electric flow storage device type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcElectricFlowStorageDeviceType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcElectricFlowStorageDeviceType** are represented by instances of *IfcElectricFlowStorageDevice*. Refer to the documentation at *IfcElectricFlowStorageDevice* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)**ENTITY** IfcElectricFlowStorageDeviceType**SUBTYPE OF** (IfcFlowStorageDeviceType);

PredefinedType : IfcElectricFlowStorageDeviceTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;[Attribute Definitions:](#)

PredefinedType : Identifies the predefined types of electric flow storage devices from which the type required may be set.

[Inheritance Graph:](#)

ENTITY IfcElectricFlowStorageDeviceType**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcFlowStorageDeviceType

ENTITY IfcElectricFlowStorageDeviceType

PredefinedType : IfcElectricFlowStorageDeviceTypeEnum;

END_ENTITY;

```
<xs:element name="IfcElectricFlowStorageDeviceType" type="Ifc: IfcElectricFlowStorageDeviceType"
substitutionGroup="Ifc: IfcFlowStorageDeviceType" nillable="true"/>
<xs:complexType name="IfcElectricFlowStorageDeviceType">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcFlowStorageDeviceType">
      <xs:attribute name="PredefinedType" type="Ifc: IfcElectricFlowStorageDeviceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricGenerator

An electric generator is an engine that is a machine for converting mechanical energy into electrical energy.

HISTORY [New entity in IFC4](#)

[EXPRESS Specification:](#)

ENTITY IfcElectricGenerator

SUBTYPE OF (IfcEnergyConversionDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcElectricGeneratorTypeEnum.USERDEFINED) OR ((PredefinedType =
IfcElectricGeneratorTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCELECTRICGENERATOR' IN
TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

[Formal Propositions:](#)

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcElectricGeneratorType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no electric generator type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcElectricGeneratorType.

[Inheritance Graph:](#)

ENTITY IfcElectricGenerator**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcEnergyConversionDevice**ENTITY** IfcElectricGenerator

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcElectricGenerator" type="ifc:IfcElectricGenerator" substitutionGroup="ifc:IfcEnergyConversionDevice"
nillable="true"/>
<xs:complexType name="IfcElectricGenerator">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricGeneratorType

The energy conversion device type **IfcElectricGeneratorType** defines commonly shared information for occurrences of electric generators. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a electric generator type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcElectricGeneratorType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcElectricGeneratorType** are represented by instances of *IfcElectricGenerator*. Refer to the documentation at *IfcElectricGenerator* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcElectricGeneratorType

SUBTYPE OF (IfcEnergyConversionDeviceType);

PredefinedType : IfcElectricGeneratorTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcElectricGeneratorTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricGeneratorTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

[Attribute Definitions:](#)

PredefinedType : Identifies the predefined types of electric generators from which the type required may be set.

[Inheritance Graph:](#)

ENTITY IfcElectricGeneratorType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] OF **UNIQUE** IfcStrippedOptional;

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcEnergyConversionDeviceType**ENTITY** IfcElectricGeneratorType

PredefinedType : IfcElectricGeneratorTypeEnum;

END_ENTITY;

```
<xs:element name="IfcElectricGeneratorType" type="ifc:IfcElectricGeneratorType"
substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/>
<xs:complexType name="IfcElectricGeneratorType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDeviceType">
      <xs:attribute name="PredefinedType" type="ifc:IfcElectricGeneratorTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricMotor

An electric motor is an engine that is a machine for converting electrical energy into mechanical energy.

HISTORY [New entity in IFC4](#)

[EXPRESS Specification:](#)**ENTITY** IfcElectricMotor

SUBTYPE OF (IfcEnergyConversionDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcElectricMotorTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricMotorTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCELECTRICMOTORTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcElectricMotorType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no electric motor type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcElectricMotorType*.

Inheritance Graph:

ENTITY IfcElectricMotor

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcEnergyConversionDevice

ENTITY IfcElectricMotor

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

<xs:element name="IfcElectricMotor" type="ifc: IfcElectricMotor" substitutionGroup="ifc: IfcEnergyConversionDevice"
 nillable="true"/>

```
<xs:complexType name="IfcElectricMotor">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcEnergyConversionDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcElectricMotorType

The energy conversion device type **IfcElectricMotorType** defines commonly shared information for occurrences of electric motors. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a electric motor type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcElectricMotorType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcElectricMotorType** are represented by instances of *IfcElectricMotor*. Refer to the documentation at *IfcElectricMotor* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcElectricMotorType

SUBTYPE OF (IfcEnergyConversionDeviceType);

PredefinedType : IfcElectricMotorTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcElectricMotorTypeEnum.USERDEFINED) OR ((PredefinedType = IfcElectricMotorTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of electric motor from which the type required may be set.

Inheritance Graph:

ENTITY IfcElectricMotorType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcEnergyConversionDeviceType**ENTITY** IfcElectricMotorType

PredefinedType : IfcElectricMotorTypeEnum;

END_ENTITY;

```

<xs:element name="IfcElectricMotorType" type="ifc: IfcElectricMotorType"
substitutionGroup="ifc: IfcEnergyConversionDeviceType" nillable="true"/>
<xs:complexType name="IfcElectricMotorType">
  <xs:complexContent>
    <xs:extension base="ifc: IfcEnergyConversionDeviceType">
      <xs:attribute name="PredefinedType" type="ifc: IfcElectricMotorTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

IfcJunctionBox

A junction box is an enclosure within which cables are connected.

Cables may be members of an electrical circuit (for electrical power systems) or be information carriers (in a telecommunications system). A junction box is typically intended to conceal a cable junction from sight, eliminate tampering or provide a safe place for electrical connection.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcJunctionBox

SUBTYPE OF (IfcFlowFitting);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcJunctionBoxTypeEnum.USERDEFINED) OR ((PredefinedType = IfcJunctionBoxTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCJUNCTIONBOXTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcJunctionBoxType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no junction box type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcJunctionBoxType*.

Inheritance Graph:

ENTITY IfcJunctionBox

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE**ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowFitting**ENTITY** IfcJunctionBoxPredefinedType : **OPTIONAL** IfcStrippedOptional;**END_ENTITY;**

```

<xs:element name="IfcJunctionBox" type="ifc: IfcJunctionBox" substitutionGroup="ifc: IfcFlowFitting" nillable="true"/>
<xs:complexType name="IfcJunctionBox">
  <xs:complexContent>
    <xs:extension base="ifc: IfcFlowFitting"/>
  </xs:complexContent>
</xs:complexType>

```

IfcJunctionBoxType

The flow fitting type **IfcJunctionBoxType** defines commonly shared information for occurrences of junction boxes. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a junction box type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcJunctionBoxType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcJunctionBoxType** are represented by instances of *IfcJunctionBox*. Refer to the documentation at *IfcJunctionBox* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)**ENTITY** IfcJunctionBoxType**SUBTYPE OF** (IfcFlowFittingType);

PredefinedType : IfcJunctionBoxTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcJunctionBoxTypeEnum.USERDEFINED) OR ((PredefinedType = IfcJunctionBoxTypeEnum.USERDEFINED) AND EXISTS(SELfIfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of junction boxes from which the type required may be set.

Inheritance Graph:

ENTITY IfcJunctionBoxType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] OF **UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowFittingType**ENTITY** IfcJunctionBoxType

PredefinedType : IfcJunctionBoxTypeEnum;

END_ENTITY;

```
<xs:element name="IfcJunctionBoxType" type="Ifc: IfcJunctionBoxType" substitutionGroup="Ifc: IfcFlowFittingType"
nillable="true"/>
<xs:complexType name="IfcJunctionBoxType">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcFlowFittingType">
      <xs:attribute name="PredefinedType" type="Ifc: IfcJunctionBoxTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcLamp

A lamp is an artificial light source such as a light bulb or tube.

HISTORY [New entity in IFC4](#)

[EXPRESS Specification:](#)**ENTITY** IfcLamp**SUBTYPE OF** (IfcFlowTerminal);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcLampTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcLampTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
 CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCLAMP' IN
 TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcLampType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no lamp type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcLampType*.

Inheritance Graph:**ENTITY** IfcLamp**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;

Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;

InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;

ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowTerminal

ENTITY IfcLamp

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcLamp" type="ifc:IfcLamp" substitutionGroup="ifc:IfcFlowTerminal" nillable="true"/>
<xs:complexType name="IfcLamp">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminal"/>
  </xs:complexContent>
</xs:complexType>
```

IfcLampType

The flow terminal type **IfcLampType** defines commonly shared information for occurrences of lamps. The set of shared information may include:

- common properties with shared property sets

- common representations
- common materials
- common composition of elements
- common ports

It is used to define a lamp type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcLampType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcLampType** are represented by instances of *IfcLamp*. Refer to the documentation at *IfcLamp* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcLampType

SUBTYPE OF (IfcFlowTerminalType);

PredefinedType : IfcLampTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcLampTypeEnum.USERDEFINED) OR ((PredefinedType = IfcLampTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of lamp from which the type required may be set.

Inheritance Graph:

ENTITY IfcLampType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0: 1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1: ?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0: 1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1: ?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowTerminalType**ENTITY** IfcLampType

PredefinedType : IfcLampTypeEnum;

END_ENTITY;

```
<xs:element name="IfcLampType" type="ifc:IfcLampType" substitutionGroup="ifc:IfcFlowTerminalType" nillable="true"/>
<xs:complexType name="IfcLampType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminalType">
      <xs:attribute name="PredefinedType" type="ifc:IfcLampTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcLightFixture

A light fixture is a container that is designed for the purpose of housing one or more lamps and optionally devices that control, restrict or vary their emission.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcLightFixture

SUBTYPE OF (IfcFlowTerminal);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcLightFixtureTypeEnum.USERDEFINED) OR ((PredefinedType = IfcLightFixtureTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCLIGHTFIXTURETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcLightFixtureType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no light fixture type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcLightFixtureType*.

Inheritance Graph:

ENTITY IfcLightFixture

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE**

ENTITY IfcFlowTerminal

ENTITY IfcLightFixture

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcLightFixture" type="ifc:IfcLightFixture" substitutionGroup="ifc:IfcFlowTerminal" nillable="true"/>
<xs:complexType name="IfcLightFixture">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminal"/>
  </xs:complexContent>
</xs:complexType>
```

IfcLightFixtureType

The flow terminal type **IfcLightFixtureType** defines commonly shared information for occurrences of light fixtures. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a light fixture type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcLightFixtureType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcLightFixtureType** are represented by instances of *IfcLightFixture*. Refer to the documentation at *IfcLightFixture* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcLightFixtureType

SUBTYPE OF (IfcFlowTerminalType);

PredefinedType : IfcLightFixtureTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcLightFixtureTypeEnum.USERDEFINED) OR ((PredefinedType = IfcLightFixtureTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

[Attribute Definitions:](#)

PredefinedType : Identifies the predefined types of light fixture from which the type required may be set.

[Inheritance Graph:](#)

ENTITY IfcLightFixtureType**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcFlowTerminalType

ENTITY IfcLightFixtureType

PredefinedType : IfcLightFixtureTypeEnum;

END_ENTITY;

```
<xs:element name="IfcLightFixtureType" type="Ifc: IfcLightFixtureType" substitutionGroup="Ifc: IfcFlowTerminalType"
nillable="true"/>
<xs:complexType name="IfcLightFixtureType">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcFlowTerminalType">
      <xs:attribute name="PredefinedType" type="Ifc: IfcLightFixtureTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcOutlet

An outlet is a device installed at a point to receive one or more inserted plugs for electrical power or communications.

Power outlets are commonly connected within a junction box; data outlets may be directly connected to a wall. For power outlets sharing the same circuit within a junction box, the ports should indicate the logical wiring relationship to the enclosing junction box, even though they may be physically connected to a cable going to another outlet, switch, or fixture.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcOutlet

SUBTYPE OF (IfcFlowTerminal);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcOutletTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcOutletTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
 CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCOUTLETTYPE' IN
 TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcOutletType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no outlet type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcOutletType.

Inheritance Graph:**ENTITY** IfcOutlet**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowTerminal

ENTITY IfcOutlet

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcOutlet" type="ifc:IfcOutlet" substitutionGroup="ifc:IfcFlowTerminal" nillable="true"/>
<xs:complexType name="IfcOutlet">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminal"/>
  </xs:complexContent>
</xs:complexType>
```

IfcOutletType

The flow terminal type **IfcOutletType** defines commonly shared information for occurrences of outlets. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define an outlet type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcOutletType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcOutletType** are represented by instances of *IfcOutlet*. Refer to the documentation at *IfcOutlet* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcOutletType

SUBTYPE OF (IfcFlowTerminalType);

PredefinedType : IfcOutletTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcOutletTypeEnum.USERDEFINED) OR ((PredefinedType = IfcOutletTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of outlet from which the type required may be set.

Inheritance Graph:

ENTITY IfcOutletType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] OF **UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowTerminalType**ENTITY** IfcOutletType

PredefinedType : IfcOutletTypeEnum;

END_ENTITY;

```
<xs:element name="IfcOutletType" type="ifc:IfcOutletType" substitutionGroup="ifc:IfcFlowTerminalType" nillable="true"/>
<xs:complexType name="IfcOutletType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowTerminalType">
      <xs:attribute name="PredefinedType" type="ifc:IfcOutletTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcProtectiveDevice

A protective device breaks an electrical circuit when a stated electric current that passes through it is exceeded.

A protective device provides protection against electrical current only (not as a general protective device). It may be used to represent the complete set of elements including both the tripping unit and the breaking unit that provide the protection. This may be particularly useful at earlier stages of design where the approach to breaking the electrical supply may be determined but the method of tripping may not. Alternatively, this entity may be used to specifically represent the breaking unit alone (in which case the tripping unit will also be specifically identified). This entity is specific to dedicated protective devices and excludes electrical outlets that may have circuit protection.

HISTORY New entity in IFC4

[EXPRESS Specification:](#)

ENTITY IfcProtectiveDevice**SUBTYPE OF** (IfcFlowController);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcProtectiveDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcProtectiveDeviceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCPROTECTIVEDEVICETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcProtectiveDeviceType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no protective device type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcProtectiveDeviceType.

Inheritance Graph:**ENTITY** IfcProtectiveDevice**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowController**ENTITY** IfcProtectiveDevice

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```

<xs:element name="IfcProtectiveDevice" type="Ifc:IfcProtectiveDevice" substitutionGroup="Ifc:IfcFlowController"
nillable="true"/>
<xs:complexType name="IfcProtectiveDevice">
  <xs:complexContent>
    <xs:extension base="Ifc:IfcFlowController"/>
  </xs:complexContent>
</xs:complexType>

```

IfcProtectiveDeviceType

The flow controller type **IfcProtectiveDeviceType** defines commonly shared information for occurrences of protective devices. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a protective device type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcProtectiveDeviceType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcProtectiveDeviceType** are represented by instances of *IfcProtectiveDevice*. Refer to the documentation at *IfcProtectiveDevice* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcProtectiveDeviceType

SUBTYPE OF (IfcFlowControllerType);

PredefinedType : IfcProtectiveDeviceTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcProtectiveDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcProtectiveDeviceTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;[Attribute Definitions:](#)

PredefinedType : Identifies the predefined types of protective device from which the type required may be set.

[Inheritance Graph:](#)

ENTITY IfcProtectiveDeviceType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowControllerType**ENTITY** IfcProtectiveDeviceType

PredefinedType : IfcProtectiveDeviceTypeEnum;

END_ENTITY;

```
<xs:element name="IfcProtectiveDeviceType" type="ifc: IfcProtectiveDeviceType" substitutionGroup="ifc: IfcFlowControllerType"
nillable="true"/>
<xs:complexType name="IfcProtectiveDeviceType">
  <xs:complexContent>
    <xs:extension base="ifc: IfcFlowControllerType">
      <xs:attribute name="PredefinedType" type="ifc: IfcProtectiveDeviceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcSolarDevice

A solar device converts solar radiation into other energy such as electric current or thermal energy.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcSolarDevice

SUBTYPE OF (IfcEnergyConversionDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcSolarDeviceTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcSolarDeviceTypeEnum.USERDEFINED) AND EXISTS
 (SELF\IfcObject.ObjectType));
 CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCSOLARDEVICETYPE' IN
 TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Inheritance Graph:

ENTITY IfcSolarDevice

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE**

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcEnergyConversionDevice

ENTITY IfcSolarDevice

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcSolarDevice" type="ifc:IfcSolarDevice" substitutionGroup="ifc:IfcEnergyConversionDevice"
nillable="true"/>
<xs:complexType name="IfcSolarDevice">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcSolarDeviceType

The energy conversion device type **IfcSolarDeviceType** defines commonly shared information for occurrences of solar devices. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a solar device type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcSolarDeviceType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcSolarDeviceType** are represented by instances of *IfcSolarDevice*. Refer to the documentation at *IfcSolarDevice* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcSolarDeviceType

SUBTYPE OF (IfcEnergyConversionDeviceType);

PredefinedType : IfcSolarDeviceTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcSolarDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSolarDeviceTypeEnum.USERDEFINED) AND EXISTS(SELFIIfcElementType.ElementType));

END_ENTITY;

Inheritance Graph:

ENTITY IfcSolarDeviceType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcEnergyConversionDeviceType

ENTITY IfcSolarDeviceType

PredefinedType : IfcSolarDeviceTypeEnum;

END_ENTITY;

```
<xs:element name="IfcSolarDeviceType" type="ifc: IfcSolarDeviceType" substitutionGroup="ifc: IfcEnergyConversionDeviceType"
nillable="true"/>
<xs:complexType name="IfcSolarDeviceType">
  <xs:complexContent>
    <xs:extension base="ifc: IfcEnergyConversionDeviceType">
      <xs:attribute name="PredefinedType" type="ifc: IfcSolarDeviceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcSwitchingDevice

A switch is used in a cable distribution system (electrical circuit) to control or modulate the flow of electricity.

Switches include those used for electrical power, communications, audio-visual, or other distribution system types as determined by the available ports.

HISTORY [New entity in IFC4](#)

[EXPRESS Specification:](#)

ENTITY IfcSwitchingDevice

SUBTYPE OF (IfcFlowController);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcSwitchingDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSwitchingDeviceTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCSWITCHINGDEVICETYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

[Formal Propositions:](#)

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcSwitchingDeviceType is associated),

CorrectTypeAssigned : or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.
 : Either there is no switching device type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcSwitchingDeviceType*.

Inheritance Graph:

ENTITY IfcSwitchingDevice

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowController**ENTITY** IfcSwitchingDevice

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcSwitchingDevice" type="ifc:IfcSwitchingDevice" substitutionGroup="ifc:IfcFlowController"
nillable="true"/>
<xs:complexType name="IfcSwitchingDevice">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowController"/>
  </xs:complexContent>
</xs:complexType>
```

IfcSwitchingDeviceType

The flow controller type **IfcSwitchingDeviceType** defines commonly shared information for occurrences of switching devices. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a switching device type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcSwitchingDeviceType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcSwitchingDeviceType** are represented by instances of *IfcSwitchingDevice*. Refer to the documentation at *IfcSwitchingDevice* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcSwitchingDeviceType

SUBTYPE OF (IfcFlowControllerType);

PredefinedType : IfcSwitchingDeviceTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcSwitchingDeviceTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSwitchingDeviceTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Identifies the predefined types of switch from which the type required may be set.

Inheritance Graph:

ENTITY IfcSwitchingDeviceType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] **OF** IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowControllerType**ENTITY** IfcSwitchingDeviceType

PredefinedType : IfcSwitchingDeviceTypeEnum;

END_ENTITY;

```
<xs:element name="IfcSwitchingDeviceType" type="ifc:IfcSwitchingDeviceType" substitutionGroup="ifc:IfcFlowControllerType"
nillable="true"/>
<xs:complexType name="IfcSwitchingDeviceType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowControllerType">
      <xs:attribute name="PredefinedType" type="ifc:IfcSwitchingDeviceTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcTransformer

A transformer is an inductive stationary device that transfers electrical energy from one circuit to another.

IfcTransformer is used to transform electric power; conversion of electric signals for other purposes is handled at other entities: *IfcController* converts arbitrary signals, *IfcAudioVisualAppliance* converts signals for audio or video streams, and *IfcCommunicationsAppliance* converts signals for data or other communications usage.

HISTORY New entity in IFC4

EXPRESS Specification:**ENTITY** IfcTransformer**SUBTYPE OF** (IfcEnergyConversionDevice);PredefinedType : **OPTIONAL** IfcStrippedOptional;**WHERE**

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcTransformerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTransformerTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCELECTRICDOMAIN.IFCTRANSFORMERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcTransformerType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no transformer type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcTransformerType*.

Inheritance Graph:**ENTITY** IfcTransformer**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcEnergyConversionDevice

ENTITY IfcTransformer

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcTransformer" type="ifc:IfcTransformer" substitutionGroup="ifc:IfcEnergyConversionDevice"
nillable="true"/>
<xs:complexType name="IfcTransformer">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcTransformerType

The energy conversion device type **IfcTransformerType** defines commonly shared information for occurrences of transformers. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a transformer type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcTransformerType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcTransformerType** are represented by instances of *IfcTransformer*. Refer to the documentation at *IfcTransformer* for supported property sets, materials, composition, and ports.

EXPRESS Specification:**ENTITY** IfcTransformerType

SUBTYPE OF (IfcEnergyConversionDeviceType);

PredefinedType : IfcTransformerTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcTransformerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTransformerTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;**Attribute Definitions:**

PredefinedType : Identifies the predefined types of transformer from which the type required may be set.

Inheritance Graph:**ENTITY** IfcTransformerType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType

ENTITY IfcEnergyConversionDeviceType

ENTITY IfcTransformerType

PredefinedType : IfcTransformerTypeEnum;

END_ENTITY;

```
<xs:element name="IfcTransformerType" type="ifc:IfcTransformerType"
substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/>
<xs:complexType name="IfcTransformerType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDeviceType">
      <xs:attribute name="PredefinedType" type="ifc:IfcTransformerTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcChiller

A chiller is a device used to remove heat from a liquid via a vapor-compression or absorption refrigeration cycle to cool a fluid, typically water or a mixture of water and glycol. The chilled fluid is then used to cool and dehumidify air in a building.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcChiller

SUBTYPE OF (IfcEnergyConversionDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcChillerTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcChillerTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
 CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCCHILLERTYPE' IN
 TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcChillerType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.
CorrectTypeAssigned : Either there is no chiller type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcChillerType.

Inheritance Graph:

ENTITY IfcChiller

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0: 1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0: 1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0: 1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0: 1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0: 1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement

INVERSE

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcEnergyConversionDevice

ENTITY IfcChiller

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcChiller" type="Ifc:IfcChiller" substitutionGroup="Ifc:IfcEnergyConversionDevice" nillable="true"/>
<xs:complexType name="IfcChiller">
  <xs:complexContent>
    <xs:extension base="Ifc:IfcEnergyConversionDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcChillerType

The energy conversion device type **IfcChillerType** defines commonly shared information for occurrences of chillers. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a chiller type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcChillerType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcChillerType** are represented by instances of *IfcChiller*. Refer to the documentation at *IfcChiller* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcChillerType

SUBTYPE OF (IfcEnergyConversionDeviceType);

PredefinedType : IfcChillerTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcChillerTypeEnum.USERDEFINED) OR ((PredefinedType = IfcChillerTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines the typical types of chillers (e.g., air-cooled, water-cooled, etc.).

Inheritance Graph:

ENTITY IfcChillerType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcEnergyConversionDeviceType**ENTITY** IfcChillerType

PredefinedType : IfcChillerTypeEnum;

END_ENTITY;

```
<xs:element name="IfcChillerType" type="ifc:IfcChillerType" substitutionGroup="ifc:IfcEnergyConversionDeviceType"
nillable="true"/>
<xs:complexType name="IfcChillerType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDeviceType">
      <xs:attribute name="PredefinedType" type="ifc:IfcChillerTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcFlowMeter

A flow meter is a device that is used to measure the flow rate in a system.

HISTORY [New entity in IFC4](#)

[EXPRESS Specification:](#)**ENTITY** IfcFlowMeter

SUBTYPE OF (IfcFlowController);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcFlowMeterTypeEnum.USERDEFINED) OR ((PredefinedType = IfcFlowMeterTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCFLOWMETERTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcFlowMeterType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no flow meter type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcFlowMeterType*.

Inheritance Graph:**ENTITY** IfcFlowMeter**ENTITY** IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition**INVERSE**

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;

Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;

IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;

HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;

IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;

Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;

HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;

Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;

IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;

IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcFlowController**ENTITY** IfcFlowMeter

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcFlowMeter" type="Ifc:IfcFlowMeter" substitutionGroup="Ifc:IfcFlowController" nillable="true"/>
<xs:complexType name="IfcFlowMeter">
  <xs:complexContent>
    <xs:extension base="Ifc:IfcFlowController"/>
  </xs:complexContent>
</xs:complexType>
```

```
</xs:complexContent>
</xs:complexType>
```

IfcFlowMeterType

The flow controller type **IfcFlowMeterType** defines commonly shared information for occurrences of flow meters. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a flow meter type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcFlowMeterType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcFlowMeterType** are represented by instances of *IfcFlowMeter*. Refer to the documentation at *IfcFlowMeter* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcFlowMeterType

SUBTYPE OF (IfcFlowControllerType);

PredefinedType : IfcFlowMeterTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcFlowMeterTypeEnum.USERDEFINED) OR ((PredefinedType = IfcFlowMeterTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines the type of flow meter.

Inheritance Graph:

ENTITY IfcFlowMeterType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType**ENTITY** IfcDistributionFlowElementType**ENTITY** IfcFlowControllerType**ENTITY** IfcFlowMeterType

PredefinedType : IfcFlowMeterTypeEnum;

END_ENTITY;

<xs:element name="IfcFlowMeterType" type="ifc: IfcFlowMeterType" substitutionGroup="ifc: IfcFlowControllerType"
 nillable="true"/>

```

<xs:complexType name="IfcFlowMeterType">
  <xs:complexContent>
    <xs:extension base="Ifc: IfcFlowControllerType">
      <xs:attribute name="PredefinedType" type="Ifc: IfcFlowMeterTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

IfcPump

A pump is a device which imparts mechanical work on fluids or slurries to move them through a channel or pipeline. A typical use of a pump is to circulate chilled water or heating hot water in a building services distribution system.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcPump

SUBTYPE OF (IfcFlowMovingDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcPumpTypeEnum.USERDEFINED) OR ((PredefinedType = IfcPumpTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCPUMPTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an IfcPumpType is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no pump type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type IfcPumpType.

Inheritance Graph:

ENTITY IfcPump

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;

OwnerHistory : **OPTIONAL** IfcOwnerHistory;

Name : **OPTIONAL** IfcLabel;

Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcObject

ObjectType : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsDeclaredBy : **SET** [0:1] OF IfcRelDefinesByObject **FOR** RelatedObjects;
 Declares : **SET OF** IfcRelDefinesByObject **FOR** RelatingObject;
 IsTypedBy : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatedObjects;
 IsDefinedBy : **SET OF** IfcRelDefinesByProperties **FOR** RelatedObjects;

ENTITY IfcProduct

ObjectPlacement : **OPTIONAL** IfcObjectPlacement;
 Representation : **OPTIONAL** IfcProductRepresentation;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE**

ENTITY IfcDistributionFlowElement

INVERSE

ENTITY IfcFlowMovingDevice

ENTITY IfcPump

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcPump" type="ifc:IfcPump" substitutionGroup="ifc:IfcFlowMovingDevice" nillable="true"/>
<xs:complexType name="IfcPump">
  <xs:complexContent>
    <xs:extension base="ifc:IfcFlowMovingDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcPumpType

The flow moving device type **IfcPumpType** defines commonly shared information for occurrences of pumps. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a pump type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcPumpType** may be declared within *IfcProject* or *IfcProjectLibrary* using *IfcRelDeclares* and may be exchanged with or without occurrences of the type. Occurrences of **IfcPumpType** are represented by instances of *IfcPump*. Refer to the documentation at *IfcPump* for supported property sets, materials, composition, and ports.

[EXPRESS Specification:](#)

ENTITY IfcPumpType

SUBTYPE OF (IfcFlowMovingDeviceType);

PredefinedType : IfcPumpTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcPumpTypeEnum.USERDEFINED) OR ((PredefinedType = IfcPumpTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : Defines the type of pump typically used in building services.

Inheritance Graph:

ENTITY IfcPumpType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
OwnerHistory : **OPTIONAL** IfcOwnerHistory;
Name : **OPTIONAL** IfcLabel;
Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
HasPropertySets : **OPTIONAL SET** [1:?] OF IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] OF IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] OF **UNIQUE** IfcStrippedOptional;
Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcFlowMovingDeviceType

ENTITY IfcPumpType

PredefinedType : IfcPumpTypeEnum;

END_ENTITY;

```
<xs:element name="IfcPumpType" type="Ifc:IfcPumpType" substitutionGroup="Ifc:IfcFlowMovingDeviceType" nillable="true"/>
<xs:complexType name="IfcPumpType">
  <xs:complexContent>
    <xs:extension base="Ifc:IfcFlowMovingDeviceType">
      <xs:attribute name="PredefinedType" type="Ifc:IfcPumpTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

IfcUnitaryEquipment

Unitary equipment typically combine a number of components into a single product, such as air handlers, pre-packaged rooftop air-conditioning units, and split systems.

HISTORY New entity in IFC4

EXPRESS Specification:

ENTITY IfcUnitaryEquipment

SUBTYPE OF (IfcEnergyConversionDevice);

PredefinedType : **OPTIONAL** IfcStrippedOptional;

WHERE

CorrectPredefinedType : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcUnitaryEquipmentTypeEnum.USERDEFINED) OR ((PredefinedType = IfcUnitaryEquipmentTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));

CorrectTypeAssigned : (SIZEOF(IsTypedBy) = 0) OR ('IFCHVACDOMAIN.IFCUNITARYEQUIPMENTTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

Formal Propositions:

CorrectPredefinedType : Either the *PredefinedType* attribute is unset (e.g. because an *IfcUnitaryEquipmentType* is associated), or the inherited attribute *ObjectType* shall be provided, if the *PredefinedType* is set to USERDEFINED.

CorrectTypeAssigned : Either there is no unitary equipment type object associated, i.e. the *IsTypedBy* inverse relationship is not provided, or the associated type object has to be of type *IfcUnitaryEquipmentType*.

Inheritance Graph:

ENTITY *IfcUnitaryEquipment*

ENTITY *IfcRoot*

GlobalId : *IfcGloballyUniqueId*;
 OwnerHistory : **OPTIONAL** *IfcOwnerHistory*;
 Name : **OPTIONAL** *IfcLabel*;
 Description : **OPTIONAL** *IfcText*;

ENTITY *IfcObjectDefinition*

INVERSE

HasAssignments : **SET OF** *IfcRelAssigns* **FOR** *RelatedObjects*;
 Nests : **SET** [0:1] OF *IfcRelNests* **FOR** *RelatedObjects*;
 IsNestedBy : **SET OF** *IfcRelNests* **FOR** *RelatingObject*;
 HasContext : **SET** [0:1] OF *IfcRelDeclares* **FOR** *RelatedDefinitions*;
 IsDecomposedBy : **SET OF** *IfcRelAggregates* **FOR** *RelatingObject*;
 Decomposes : **SET** [0:1] OF *IfcRelAggregates* **FOR** *RelatedObjects*;
 HasAssociations : **SET OF** *IfcRelAssociates* **FOR** *RelatedObjects*;

ENTITY *IfcObject*

ObjectType : **OPTIONAL** *IfcStrippedOptional*;

INVERSE

IsDeclaredBy : **SET** [0:1] OF *IfcRelDefinesByObject* **FOR** *RelatedObjects*;
 Declares : **SET OF** *IfcRelDefinesByObject* **FOR** *RelatingObject*;
 IsTypedBy : **SET** [0:1] OF *IfcRelDefinesByType* **FOR** *RelatedObjects*;
 IsDefinedBy : **SET OF** *IfcRelDefinesByProperties* **FOR** *RelatedObjects*;

ENTITY *IfcProduct*

ObjectPlacement : **OPTIONAL** *IfcObjectPlacement*;
 Representation : **OPTIONAL** *IfcProductRepresentation*;

INVERSE

ReferencedBy : **SET OF** *IfcRelAssignsToProduct* **FOR** *RelatingProduct*;

ENTITY IfcElement

Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

IsInterferedByElements : **SET OF** IfcRelInterferesElements **FOR** RelatedElement;
 InterferesElements : **SET OF** IfcRelInterferesElements **FOR** RelatingElement;
 ContainedInStructure : **SET** [0:1] OF IfcRelContainedInSpatialStructure **FOR** RelatedElements;

ENTITY IfcDistributionElement**INVERSE****ENTITY** IfcDistributionFlowElement**INVERSE****ENTITY** IfcEnergyConversionDevice**ENTITY** IfcUnitaryEquipment

PredefinedType : **OPTIONAL** IfcStrippedOptional;

END_ENTITY;

```
<xs:element name="IfcUnitaryEquipment" type="ifc:IfcUnitaryEquipment" substitutionGroup="ifc:IfcEnergyConversionDevice"
nillable="true"/>
<xs:complexType name="IfcUnitaryEquipment">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDevice"/>
  </xs:complexContent>
</xs:complexType>
```

IfcUnitaryEquipmentType

The energy conversion device type **IfcUnitaryEquipmentType** defines commonly shared information for occurrences of unitary equipments. The set of shared information may include:

- common properties with shared property sets
- common representations
- common materials
- common composition of elements
- common ports

It is used to define a unitary equipment type specification indicating the specific product information that is common to all occurrences of that product type. The **IfcUnitaryEquipmentType** may be declared within *IfcProject* or *IfcProjectLibrary* using

IfcRelDeclares and may be exchanged with or without occurrences of the type. Occurrences of **IfcUnitaryEquipmentType** are represented by instances of *IfcUnitaryEquipment*. Refer to the documentation at *IfcUnitaryEquipment* for supported property sets, materials, composition, and ports.

EXPRESS Specification:

ENTITY IfcUnitaryEquipmentType

SUBTYPE OF (IfcEnergyConversionDeviceType);

PredefinedType : IfcUnitaryEquipmentTypeEnum;

WHERE

CorrectPredefinedType : (PredefinedType <> IfcUnitaryEquipmentTypeEnum.USERDEFINED) OR ((PredefinedType = IfcUnitaryEquipmentTypeEnum.USERDEFINED) AND EXISTS(SELF\IfcElementType.ElementType));

END_ENTITY;

Attribute Definitions:

PredefinedType : The type of unitary equipment.

Inheritance Graph:

ENTITY IfcUnitaryEquipmentType

ENTITY IfcRoot

GlobalId : IfcGloballyUniqueId;
 OwnerHistory : **OPTIONAL** IfcOwnerHistory;
 Name : **OPTIONAL** IfcLabel;
 Description : **OPTIONAL** IfcText;

ENTITY IfcObjectDefinition

INVERSE

HasAssignments : **SET OF** IfcRelAssigns **FOR** RelatedObjects;
 Nests : **SET** [0:1] OF IfcRelNests **FOR** RelatedObjects;
 IsNestedBy : **SET OF** IfcRelNests **FOR** RelatingObject;
 HasContext : **SET** [0:1] OF IfcRelDeclares **FOR** RelatedDefinitions;
 IsDecomposedBy : **SET OF** IfcRelAggregates **FOR** RelatingObject;
 Decomposes : **SET** [0:1] OF IfcRelAggregates **FOR** RelatedObjects;
 HasAssociations : **SET OF** IfcRelAssociates **FOR** RelatedObjects;

ENTITY IfcTypeObject

ApplicableOccurrence : **OPTIONAL** IfcStrippedOptional;
 HasPropertySets : **OPTIONAL SET** [1:?] **OF** IfcPropertySetDefinition;

INVERSE

Types : **SET** [0:1] **OF** IfcRelDefinesByType **FOR** RelatingType;

ENTITY IfcTypeProduct

RepresentationMaps : **OPTIONAL LIST** [1:?] **OF UNIQUE** IfcStrippedOptional;
 Tag : **OPTIONAL** IfcStrippedOptional;

INVERSE

ReferencedBy : **SET OF** IfcRelAssignsToProduct **FOR** RelatingProduct;

ENTITY IfcElementType

ElementType : **OPTIONAL** IfcStrippedOptional;

ENTITY IfcDistributionElementType

ENTITY IfcDistributionFlowElementType

ENTITY IfcEnergyConversionDeviceType

ENTITY IfcUnitaryEquipmentType

PredefinedType : IfcUnitaryEquipmentTypeEnum;

END_ENTITY;

```
<xs:element name="IfcUnitaryEquipmentType" type="ifc:IfcUnitaryEquipmentType"
substitutionGroup="ifc:IfcEnergyConversionDeviceType" nillable="true"/>
<xs:complexType name="IfcUnitaryEquipmentType">
  <xs:complexContent>
    <xs:extension base="ifc:IfcEnergyConversionDeviceType">
      <xs:attribute name="PredefinedType" type="ifc:IfcUnitaryEquipmentTypeEnum" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

4.7.6.1.3 Data definition reference schema list

Each referenced schema is listed by standards body notation and official title.

Reference	Description
-----------	-------------

ISO 16739:2013	Industry Foundation Classes (IFC) for data sharing in the construction and facilities management industries
----------------	---

4.7.6.2 Concept definitions

4.7.6.2.1 Concept list

Each concept is listed by entity name and concept template within the following table. Each row corresponds to an entity, each column corresponds to a concept template, and each cell indicates usage of a concept template for an entity.

Entity	Identity	Organization Data	Actor Assignment	Revision Control	Descriptions	Spatial Decomposition	Conversion Units	Project Context	Properties for Performance	Process Assignment	Task Scheduling	Sequential Connectivity	Control Assignment	Properties for Performance	Process Declaration	Project Context	Conversion Units	Spatial Decomposition	Descriptions	Revision Control	Actor Assignment	Organization Data	Identity
IfcActor	X	X	X	X																			
IfcProject	X				X	X	X	X	X														
IfcPerformanceHistory	X								X	X													
IfcTask	X				X						X	X	X										
IfcTaskType	X																						
IfcWorkCalendar	X								X														
IfcWorkPlan	X								X														
IfcBuilding	X	X				X																	
IfcBuildingStorey	X	X				X																	
IfcDistributionElement		X																					
IfcGrid	X																						
IfcSite	X					X																	
IfcSpace	X	X																					
IfcSpaceType	X																						
IfcTransportElement	X	X																					
IfcTransportElementType	X																						

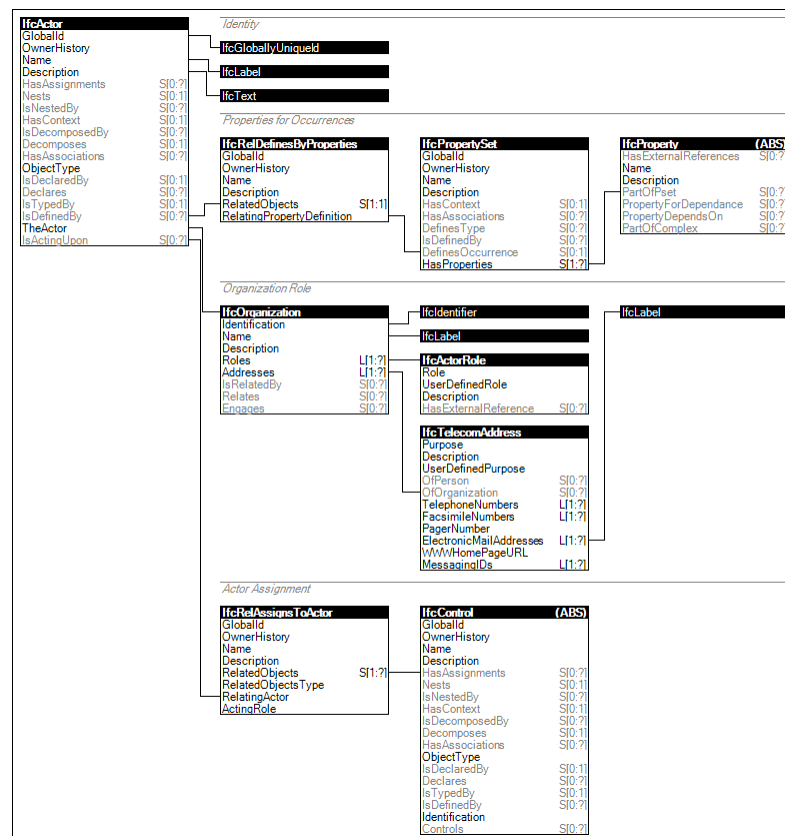
[illegible]

[illegible]

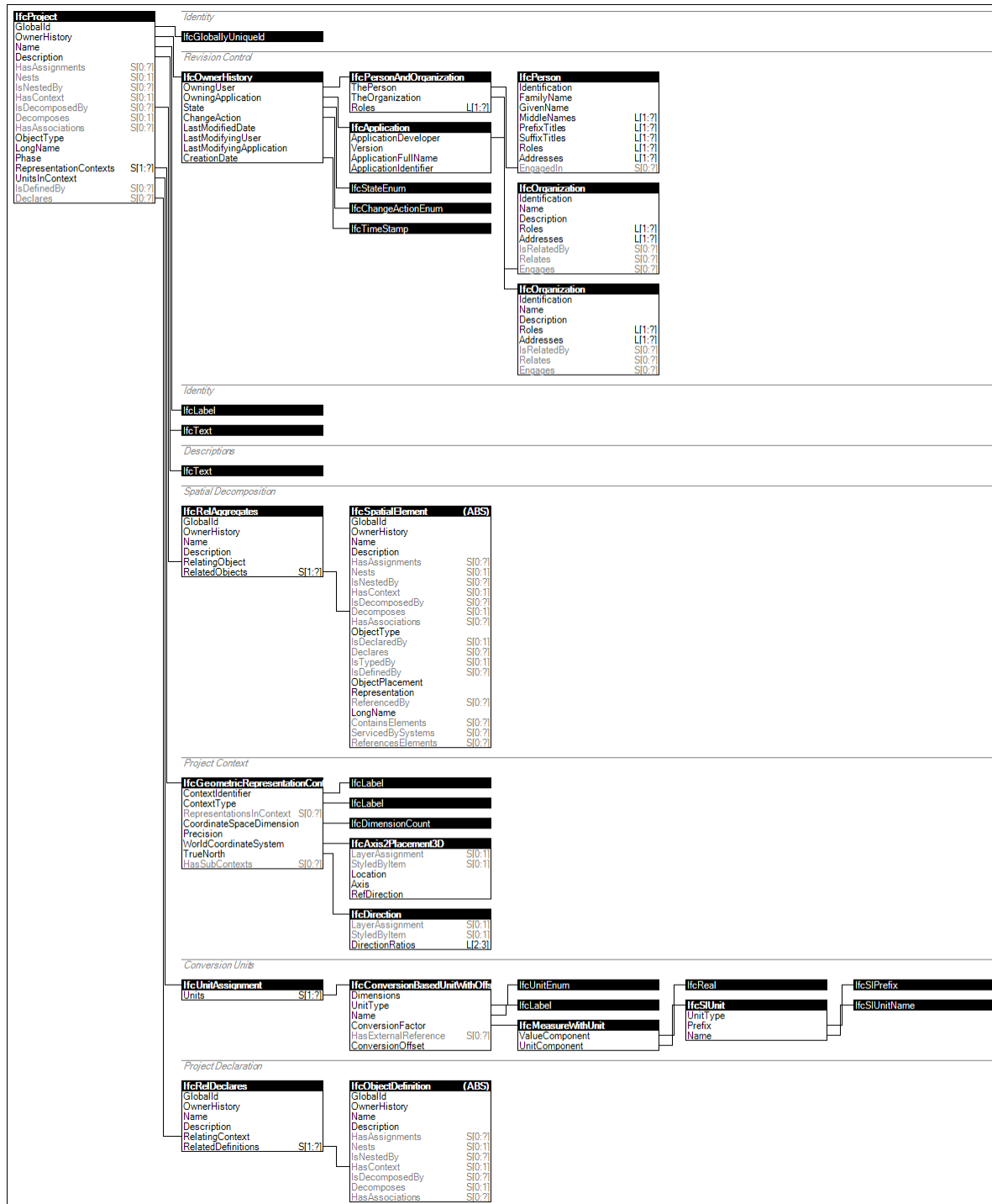
4.7.6.2.2 Concept definitions

Each entity is described within subsections, with diagrams indicating the graph of attributes and objects representing the combination of all concepts applied to instances of the entity. Each block in the diagram represents an entity, where the entity name is shown at the top of the block with background in black. Each attribute within the entity is shown in order, where black is used to indicate a direct attribute and grey is used to indicate an inverse attribute. Notation to the right of each attribute indicates aggregation, where S indicates a SET (unordered unique objects) and L indicates a LIST (ordered objects), the first number in brackets indicates the minimum count, and the second number in brackets indicates the maximum count or “?” for unlimited. Lines connecting blocks indicates attributes that point to objects of other data definitions.

IfcActor



IfcProject



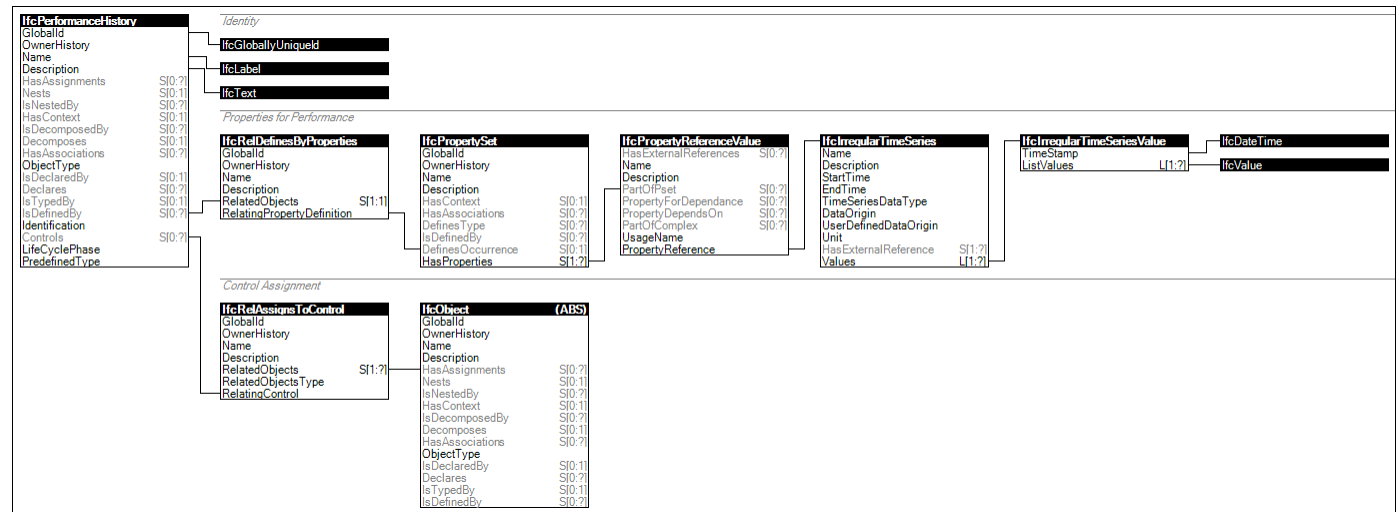
Conversion Units

If units are omitted, SI base units are implicit.

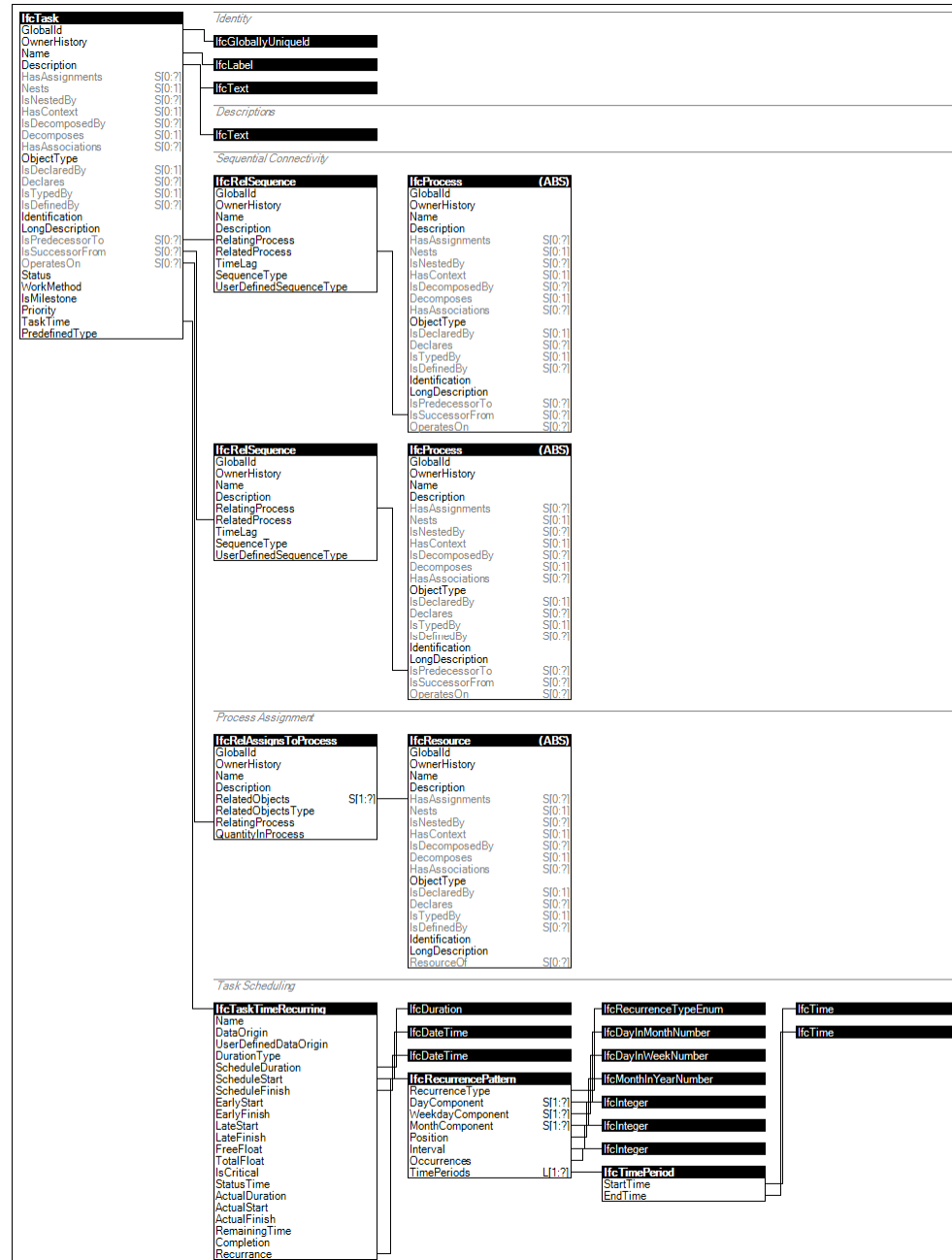
Project Declaration

Objects are declared within a project (or project library) to indicate top-level positioning within the project hierarchy and to indicate the relevant context for units and geometry.

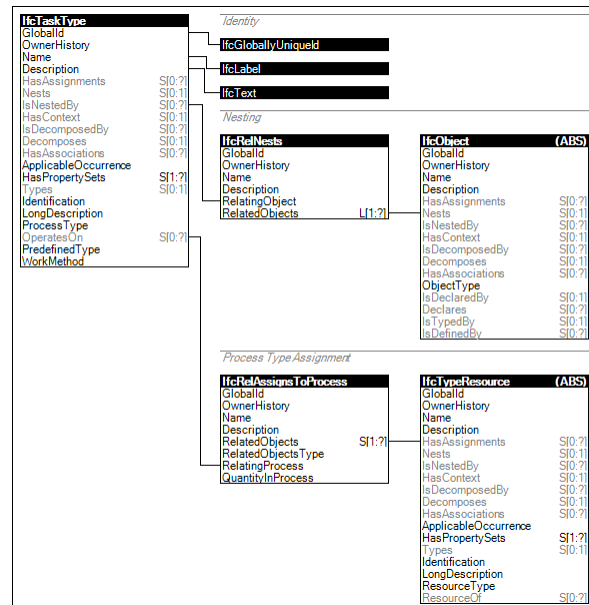
IfcPerformanceHistory



IfcTask



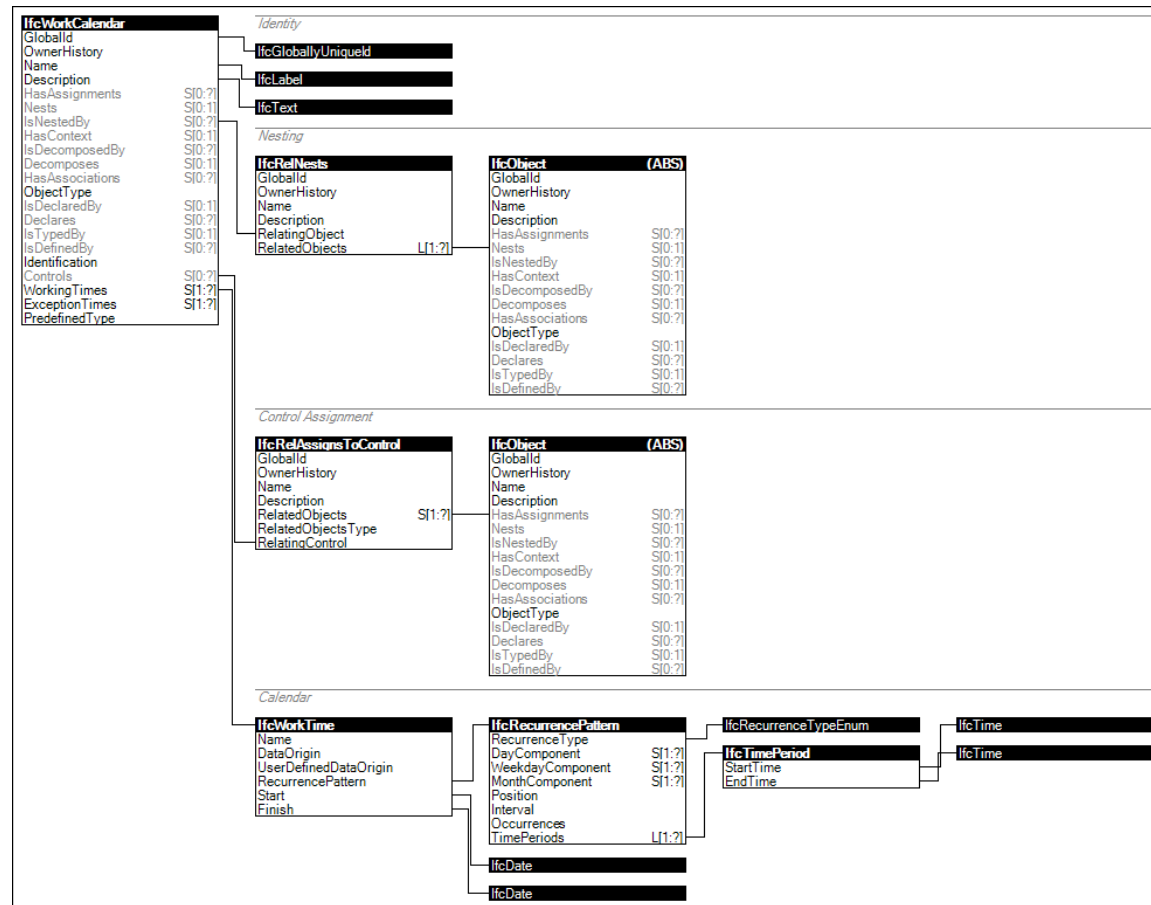
IfcTaskType



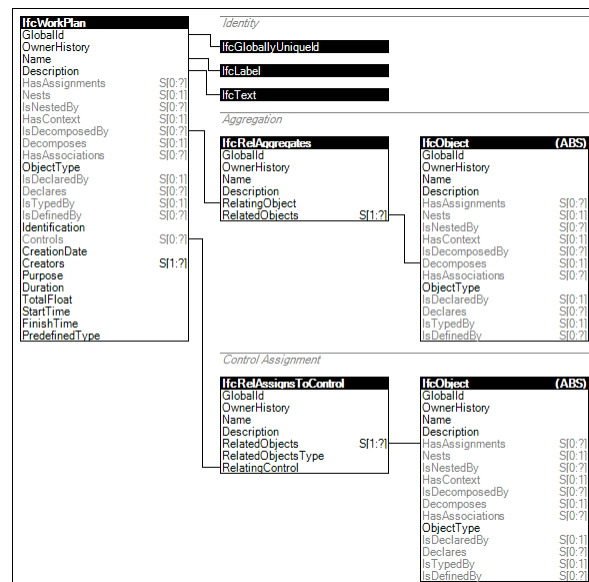
Nesting

IfcTaskType may nest other *IfcTaskType* or *IfcTask* entities using the *IfcRelNests* relationship. Such nesting indicates decomposed level of detail. Nesting of *IfcTask* entities is used if a task type shall be detailed by a sequence of tasks or if there is a need to include additional time information such as the duration of subtasks. Please note that *IfcTask* entities being contained within an *IfcTaskType* are linked with their task occurrences via *IfcRelDefinesByObject* relationships. It is also possible to define a task type for these *IfcTask* entities via *IfcRelDefinesByType* relationships. For further information please see the documentation of *IfcRelDefinesByObject*.

IfcWorkCalendar



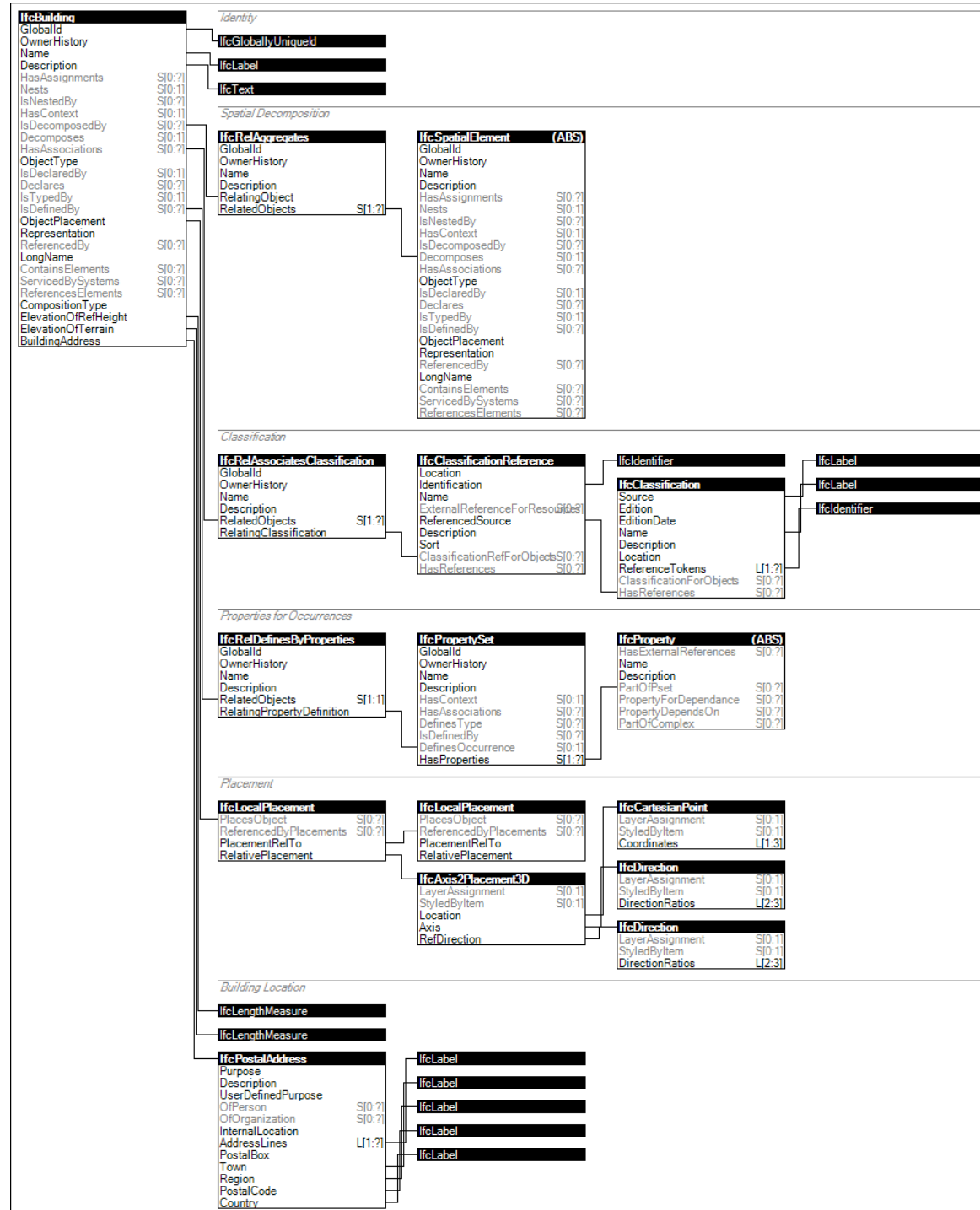
IfcWorkPlan



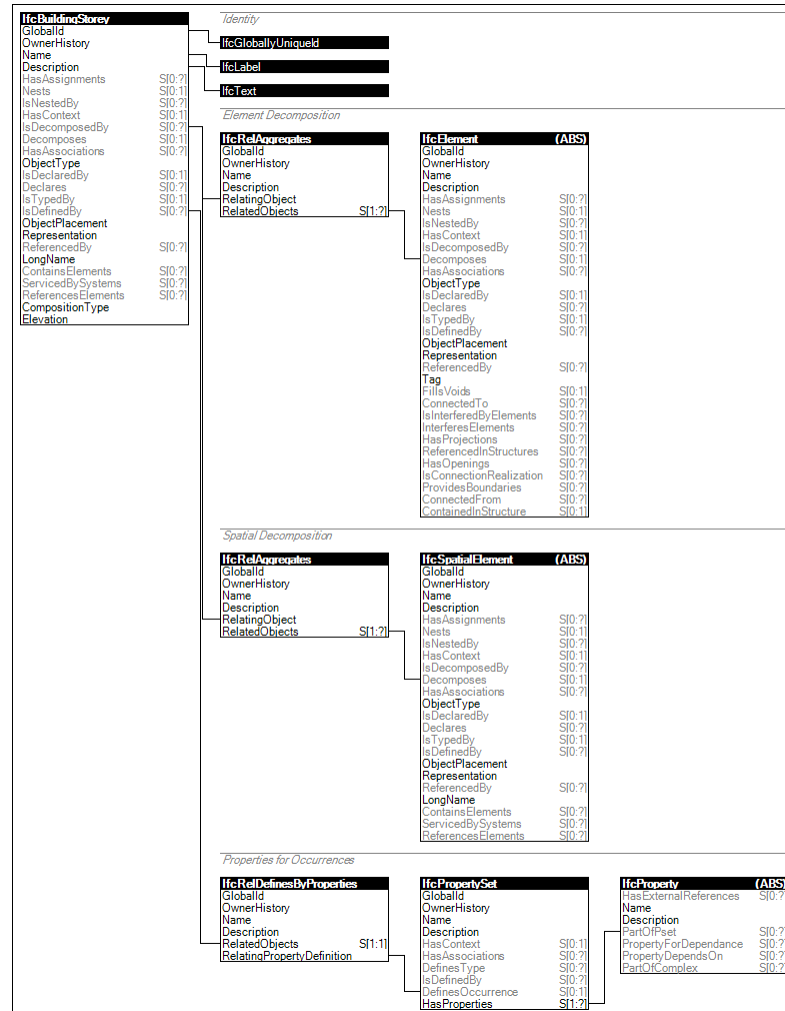
IfcBuilding

A building represents a structure that provides shelter for its occupants or contents and stands in one place.

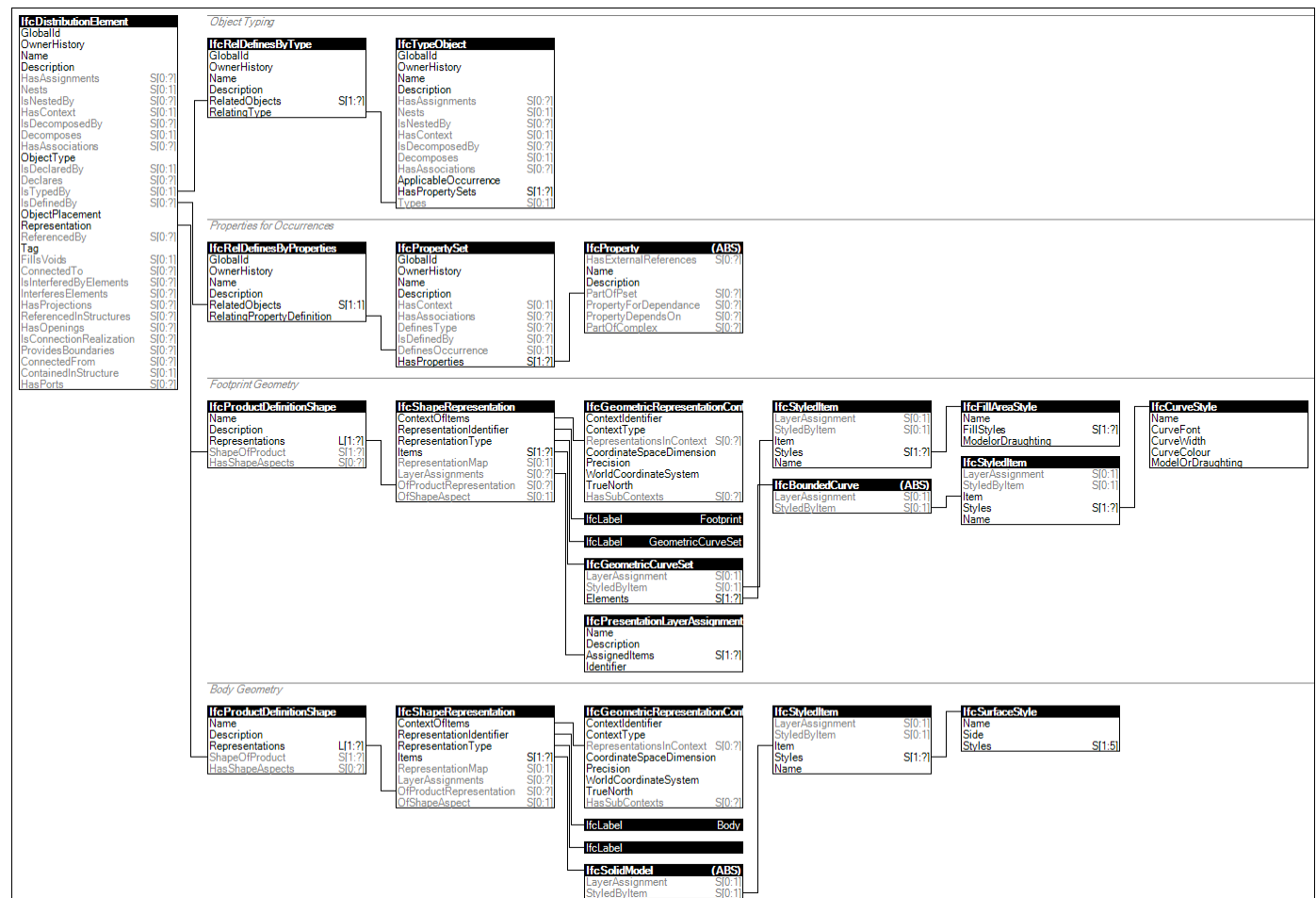
For this MVD, a building may contain variable levels of detail. During initial design, it may not contain any storeys, spaces, or containing geometry. For later design, the building is decomposed into full element detail.



IfcBuildingStorey



IfcDistributionElement



Object Typing

The *IfcDistributionElement* defines the occurrence of any HVAC, electrical, sanitary or other element within a distribution system. Common information about distribution element types (or styles) is handled by subtypes of *IfcDistributionElementType*. The *IfcDistributionElementType* (if present) may establish the common type name, usage (or predefined) type, common material, common set of properties and common shape representations (using *IfcRepresentationMap*). The *IfcDistributionElementType* is attached using the *IfcRelDefinedByType.RelatingType* objectified relationship and is accessible by the inverse *IsDefinedBy* attribute.

The assignment of types to distribution element occurrences is vital for providing the additional meaning, or ontology, of the distribution element. Many specialized type are defined in other schemas of this specification.

Footprint Geometry

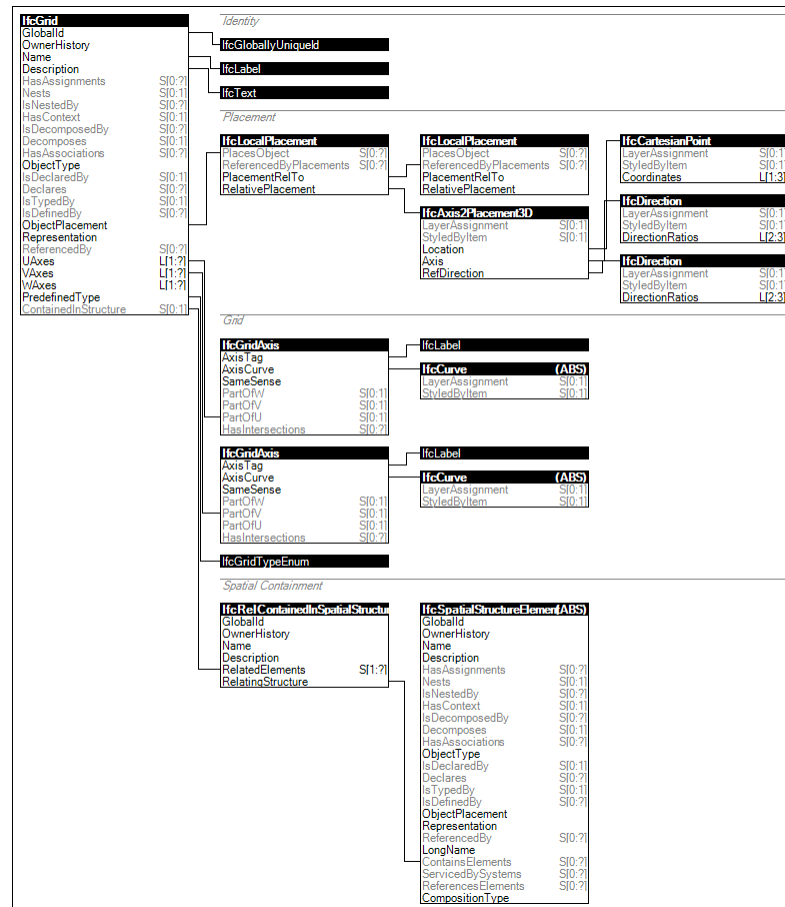
An additional representation type for all "simple" distribution flow elements (general ducts and pipes) is the ability to have a simple line based representation. The *IfcShapeRepresentation* has:

- *RepresentationIdentifier* : 'FootPrint'
- *RepresentationType* : 'GeometricCurveSet'

Body Geometry

The geometric representation of *IfcDistributionElement* is defined using different geometric representation types for the various subtypes. Only general recommendations are given at the level of the supertype, further constraints are defined at the level of its subtypes.

IfcGrid

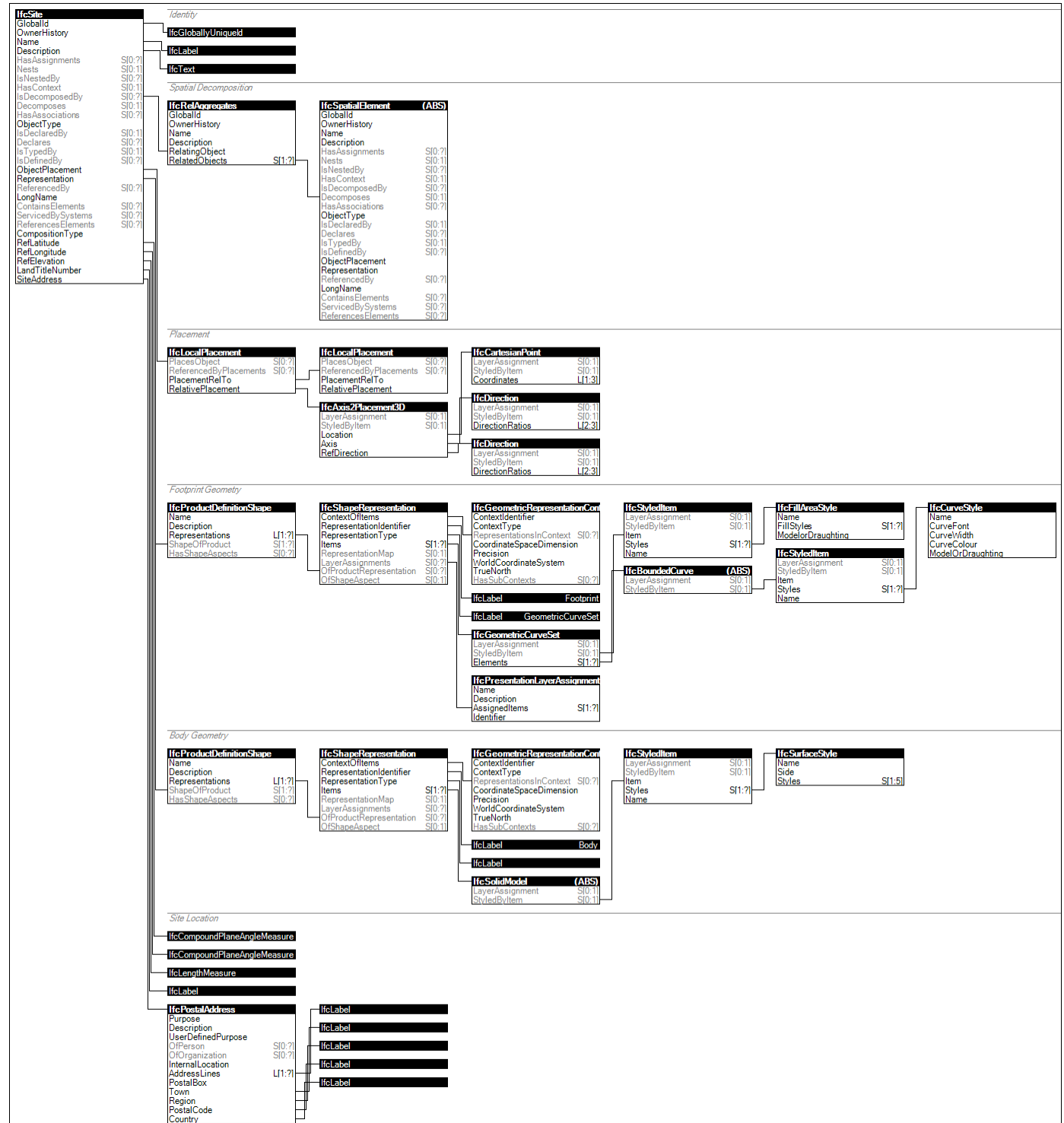


Placement

The local placement for *IfcGrid* is defined in its supertype *IfcProduct*. It is defined by the *IfcLocalPlacement*, which defines the local coordinate system that is referenced by all geometric representations.

- The *PlacementRelTo* relationship of *IfcLocalPlacement* shall point (if given) to the local placement of the same *IfcSpatialStructureElement*, which is used in the *ContainedInStructure* inverse attribute, or to a spatial structure element at a higher level, referenced by that.
- If the relative placement is not used, the absolute placement is defined within the world coordinate system.

IfcSite



Footprint Geometry

The foot print representation of *IfcSite* is given by either a single 2D curve (such as *IfcPolyline* or *IfcCompositeCurve*), or by a list of 2D curves (in case of inner boundaries).

The representation identifier and type of this geometric representation of *IfcSite* is:

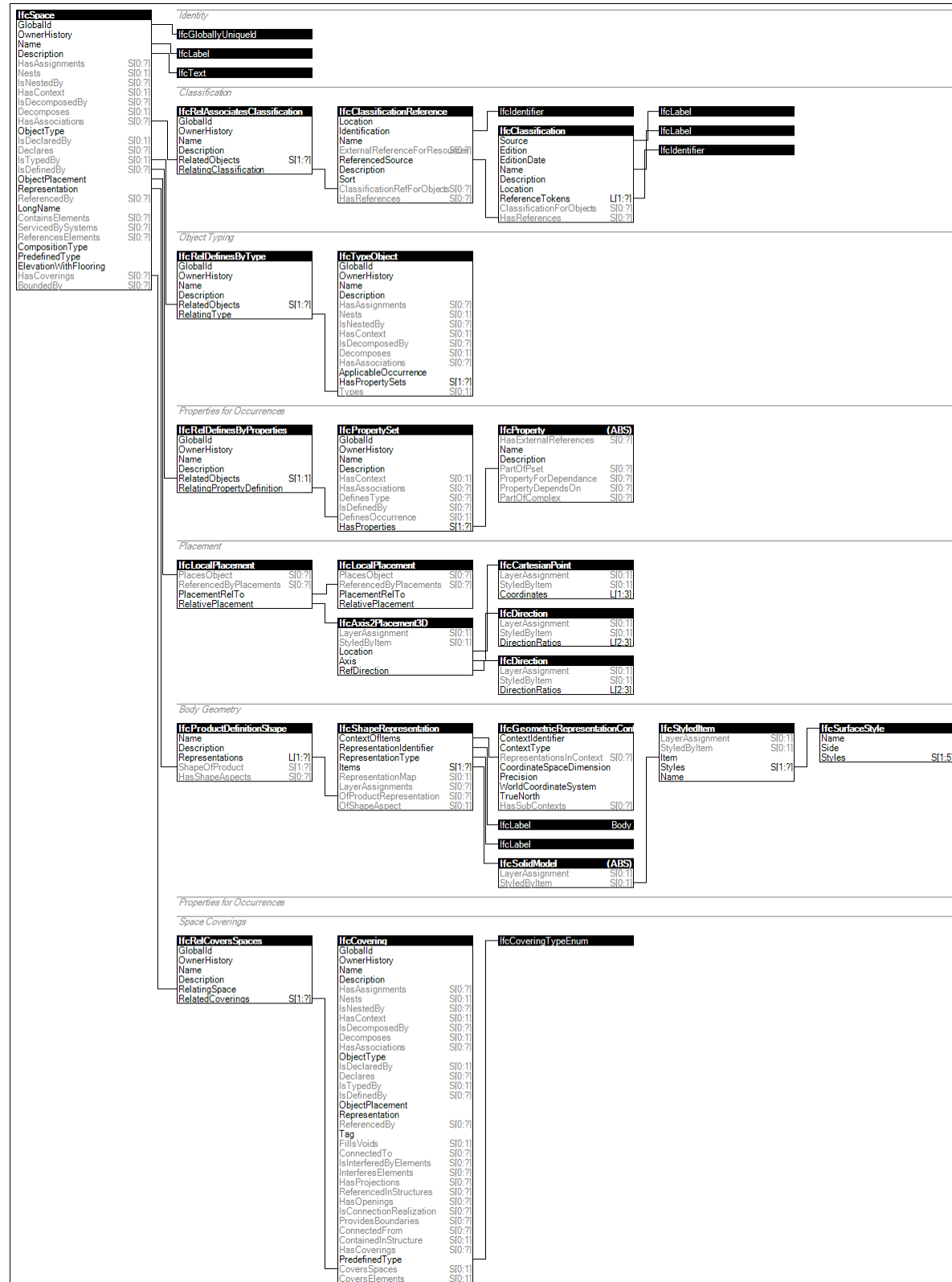
- *IfcShapeRepresentation.RepresentationIdentifier* = 'FootPrint'
- *IfcShapeRepresentation.RepresentationType* = 'GeometricCurveSet', or 'Annotation2D'

IfcSpace

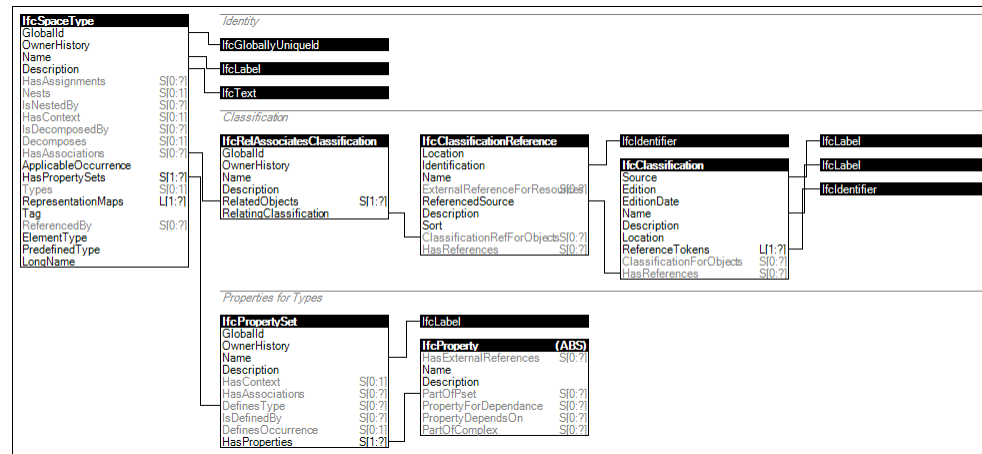
A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.

For this model view definition, the level of detail of a space may vary:

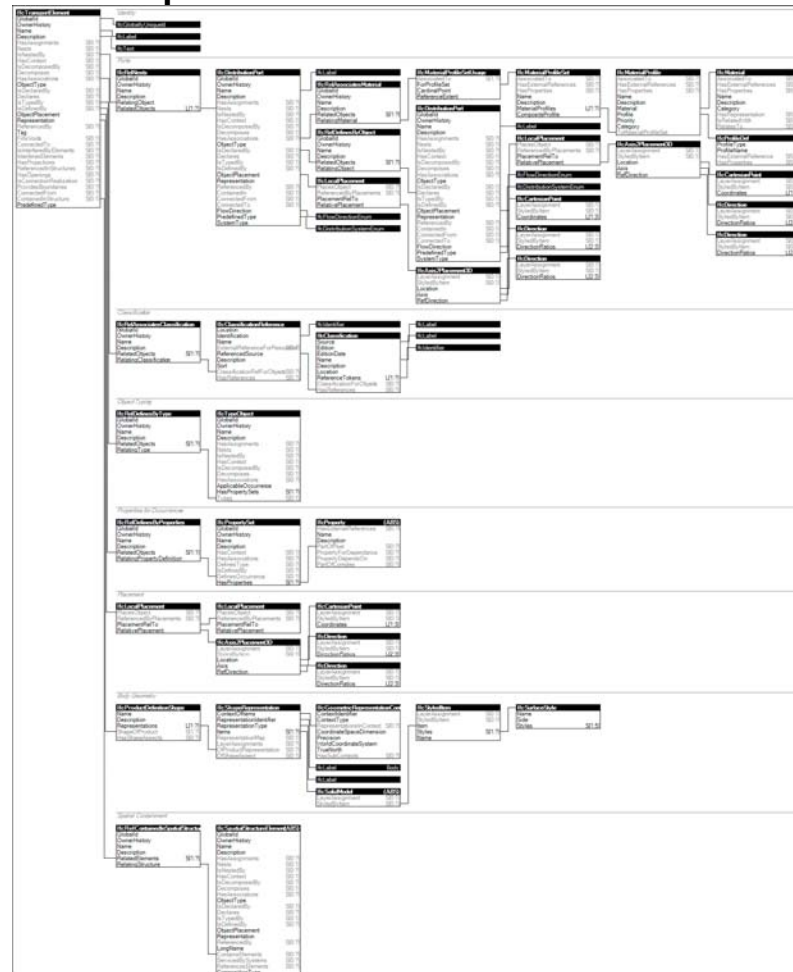
- Initially, a space may represent a count of a given space type, such as the number of small offices in a building, without having any placement or representation.
- As detailing evolves, spaces may be given standard shapes, but no placement.
- As the building takes shape, spaces may then indicate shapes, placement, and relationships to surrounding building elements.



IfcSpaceType



IfcTransportElement



Object Typing

IfcTransportElement defines the occurrence of any transportation device, common information about transportation device types (or styles) is handled by *IfcTransportElementType*. The *IfcTransportElementType* (if present) may establish the common type name, usage (or predefined) type, common material layer set, common set of properties and common shape representations (using *IfcRepresentationMap*). The *IfcTransportElementType* is attached using the *IfcRelDefinedByType.RelatingType* objectified relationship and is accessible by the inverse *IsTypedBy* attribute.

If no *IfcTransportElementType* is attached (i.e. if only occurrence information is given) the *PredefinedType* should be provided. If set to .USERDEFINED, a user defined value can be provided by the *ObjectType* attribute.

Placement

The local placement for *IfcTransportElement* is defined in its supertype *IfcProduct*. It is defined by the *IfcLocalPlacement*, which defines the local coordinate system that is referenced by all geometric representations.

- The *PlacementRelTo* relationship of *IfcLocalPlacement* shall point (if given) to the local placement of the same *IfcSpatialStructureElement*, which is used in the *ContainedInStructure* inverse attribute, or to a spatial structure element at a higher level, referenced by that.
- If the relative placement is not used, the absolute placement is defined within the world coordinate system.

Body Geometry

The body representation of any *IfcTransportElement* has the following representation types: 'Tessellation', 'SurfaceModel', 'Brep', and 'MappedRepresentation'.

NOTE Other representation types might be specified in view definitions or implementer agreements.

NOTE In addition to the 'Body' representation, an *IfcTransportElement* may have a 'Body-FallBack' representation assigned. It provides an easier to read and visualize geometric representation.

EXAMPLE In addition to a 'Body' representation of type 'Clipping', or 'CSG' and 'Body-FallBack' representation of type 'Tessellation' could be added that has all Boolean operations and curve segmentation already resolved.

Tessellation Representation Type

Any *IfcBuildingElement* (so far no further constraints are defined at the level of its subtypes) may be represented as a single or multiple tessellated surface models, including a triangulated surface model. The following attribute values for the *IfcShapeRepresentation* holding this geometric representation shall be used:

- *RepresentationIdentifier* : 'Body', or 'Body-FallBack'
- *RepresentationType* : 'Tessellation'

SurfaceModel Representation

Any *IfcTransportElement* (so far no further constraints are defined at the level of its subtypes) may be represented as a single or multiple surface models, based on either shell or face based models. Then the following attribute values for the *IfcShapeRepresentation* holding this geometric representation shall be used:

- *RepresentationIdentifier* : 'Body'
- *RepresentationType* : 'SurfaceModel'

Brep Representation

Any *IfcTransportElement* (so far no further constraints are defined at the level of its subtypes) may be represented as a single or multiple Boundary Representation elements (which are restricted to faceted Brep with or without voids). Then the following attribute values for the *IfcShapeRepresentation* holding this geometric representation shall be used:

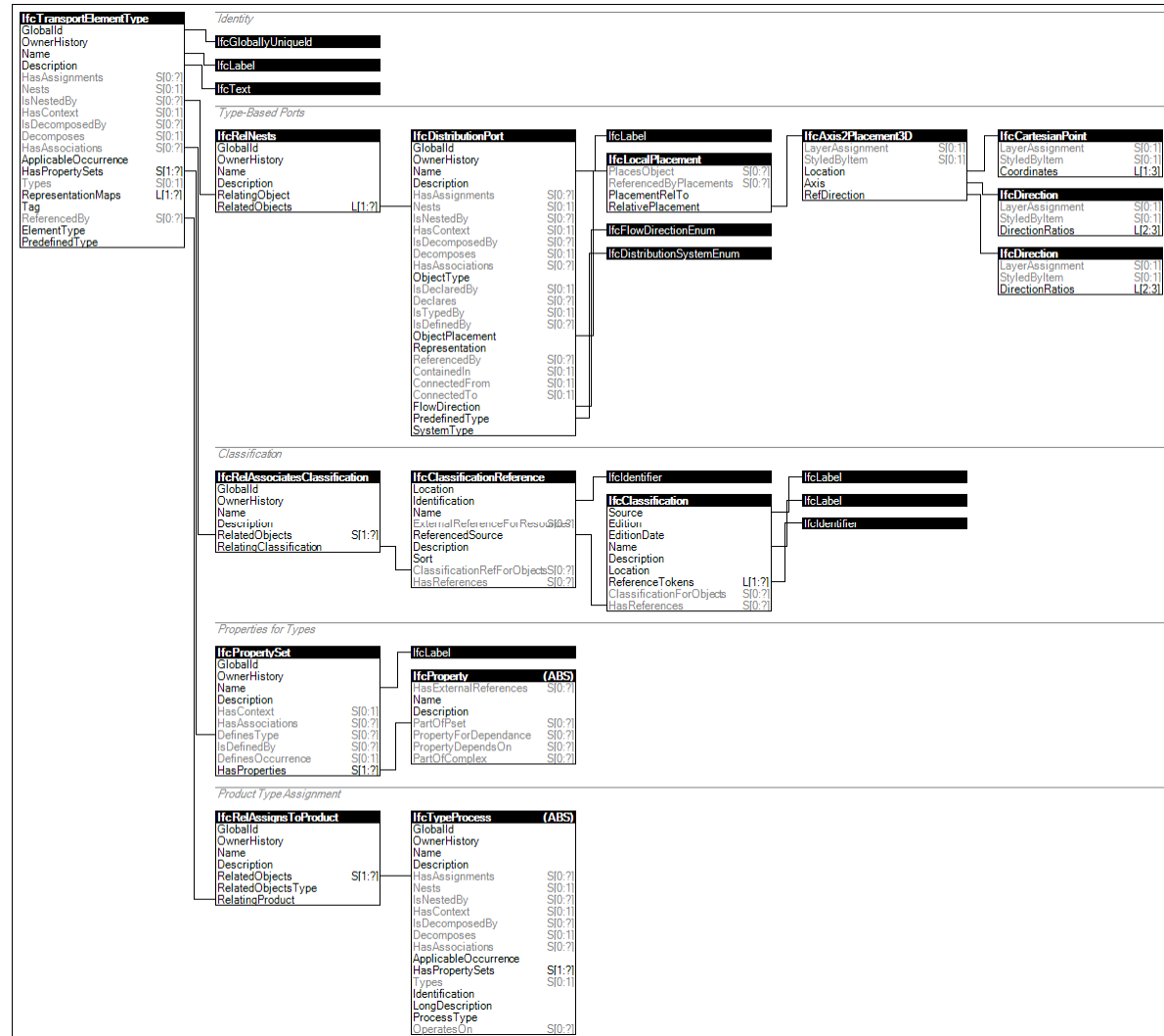
- *RepresentationIdentifier* : 'Body'
- *RepresentationType* : 'Brep'

MappedRepresentation

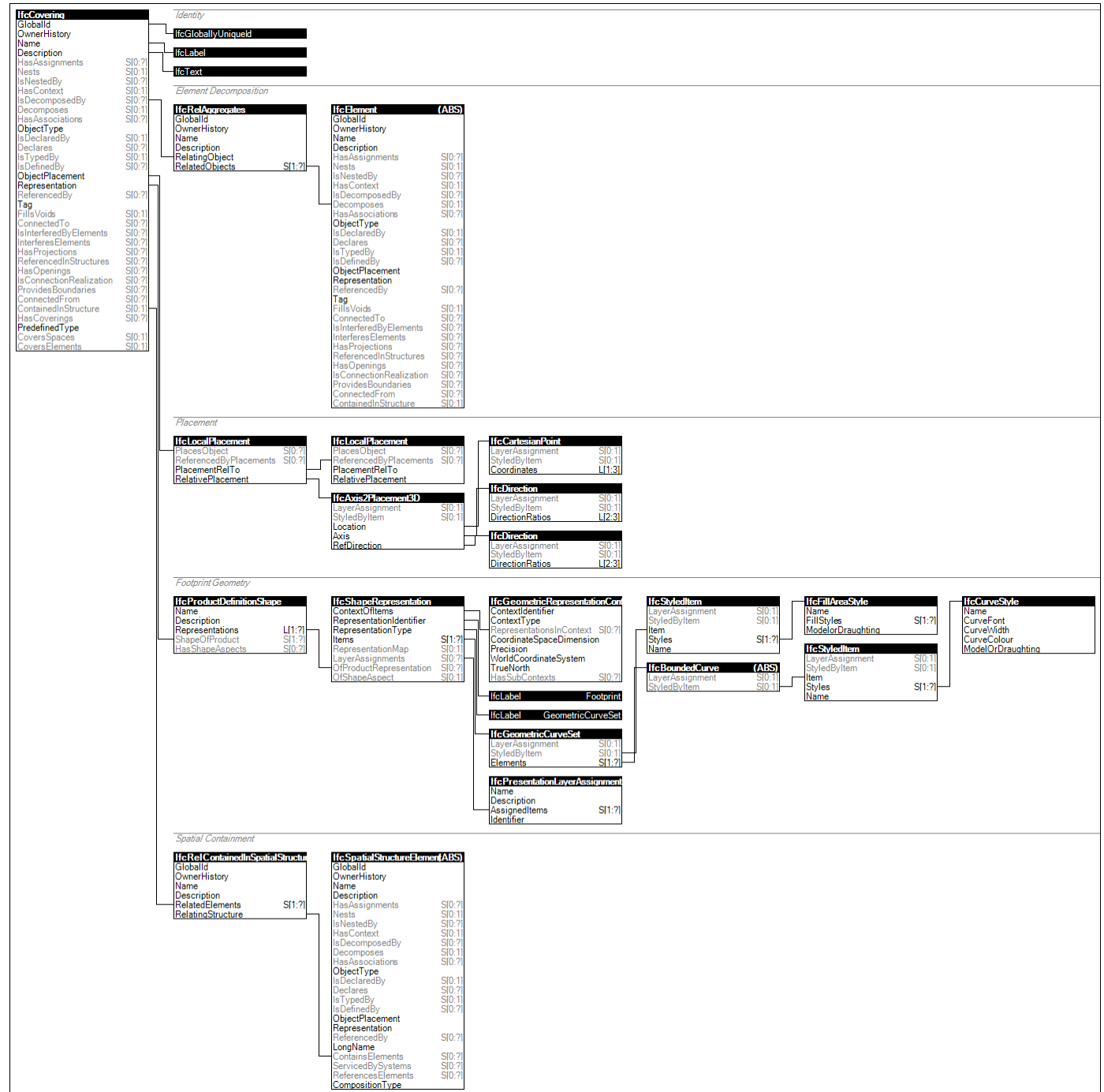
The mapped item, *IfcMappedItem*, should be used if appropriate as it allows for reusing the geometry definition of the property element type at occurrences of the same equipment type. Then the following attribute values for the *IfcShapeRepresentation* holding this geometric representation shall be used:

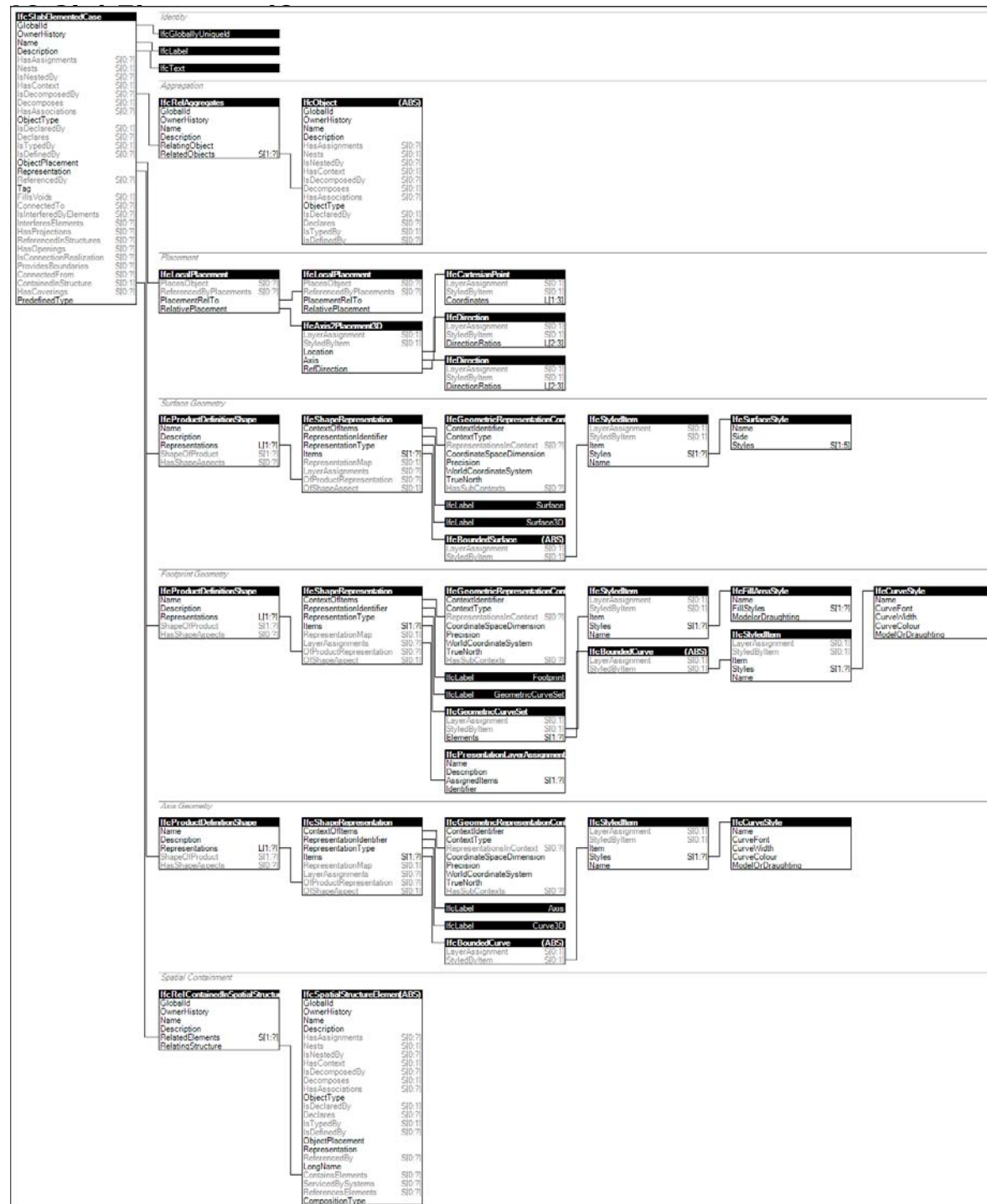
- *RepresentationIdentifier* : 'Body'
- *RepresentationType* : 'MappedRepresentation'

IfcTransportElementType



IfcCovering





Surface Geometry

The standard geometric representation of *IfcSlabElementedCase* is defined using the following multiple shape representations for its definition:

- **Surface:** A three-dimensional surface being a subtype of *IfcBoundedSurface* defining the reference surface for the elemented slab. It may be used as a simplified representation directly at the elemented slab.

NOTE It is invalid to exchange a 'Body' shape representation of an *IfcSlabElementedCase*. The body geometry is defined by the parts within the decomposition.

The surfacic geometric representation of *IfcSlabElementedCase* is defined using the 'Surface' representation.

- *RepresentationIdentifier* : 'Surface'
- *RepresentationType* : 'Surface3D'

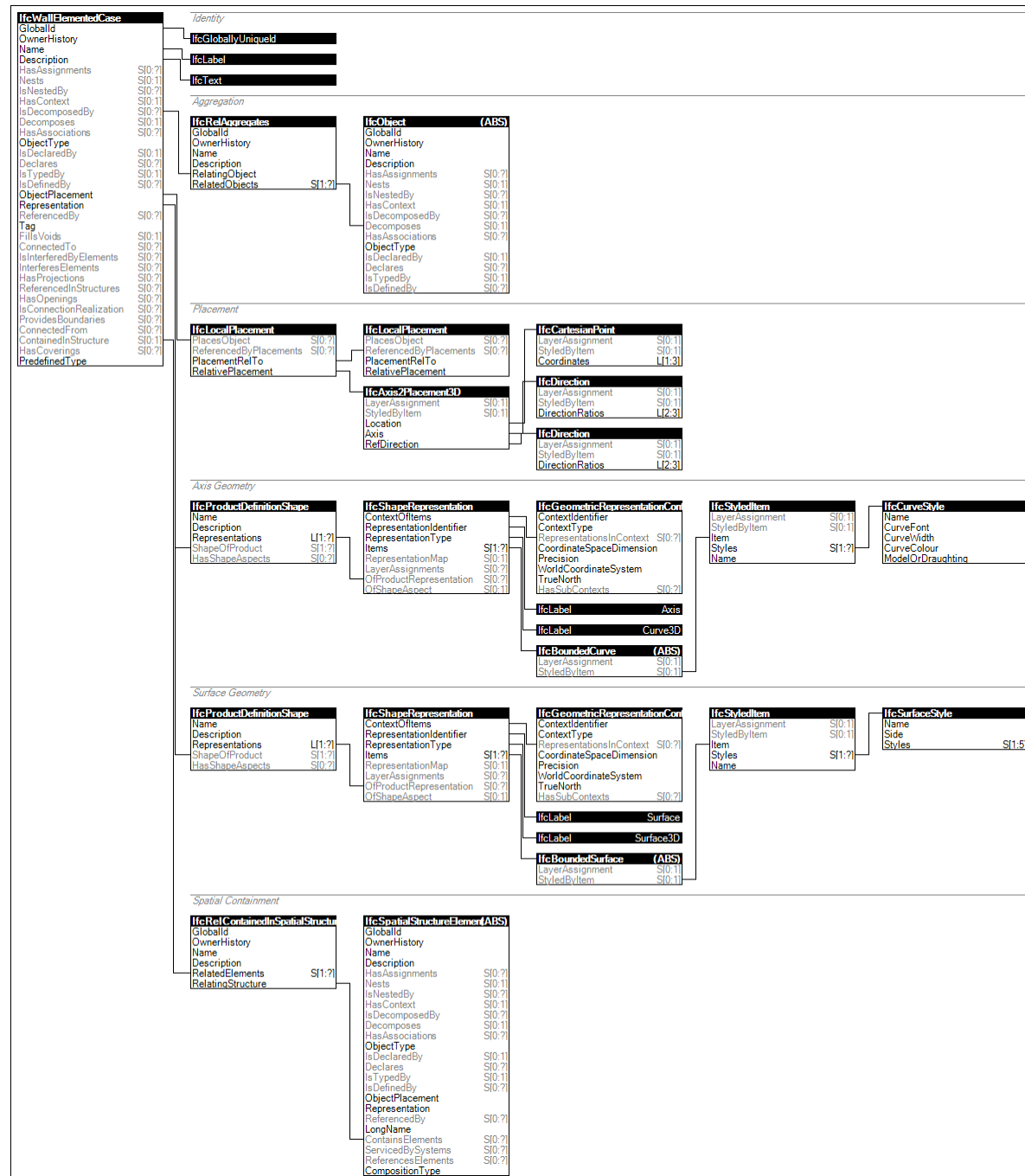
NOTE The 'Surface' can be used to define a surfacic model of the building (e.g. for analytical purposes, or for reduced Level of Detail representation). It should suppress the geometric details of the parts in the decomposition.

Aggregation

The following guidance is provided for the components of the *IfcSlabElementedCase*. The following component entity types should be used:

- Precast hollow core slabs
 - double tee or plank components : *IfcBeam*
 - topping : *IfcBuildingElementPart*
 - others : *IfcBuildingElementPart*

IfcWallElementedCase



Surface Geometry

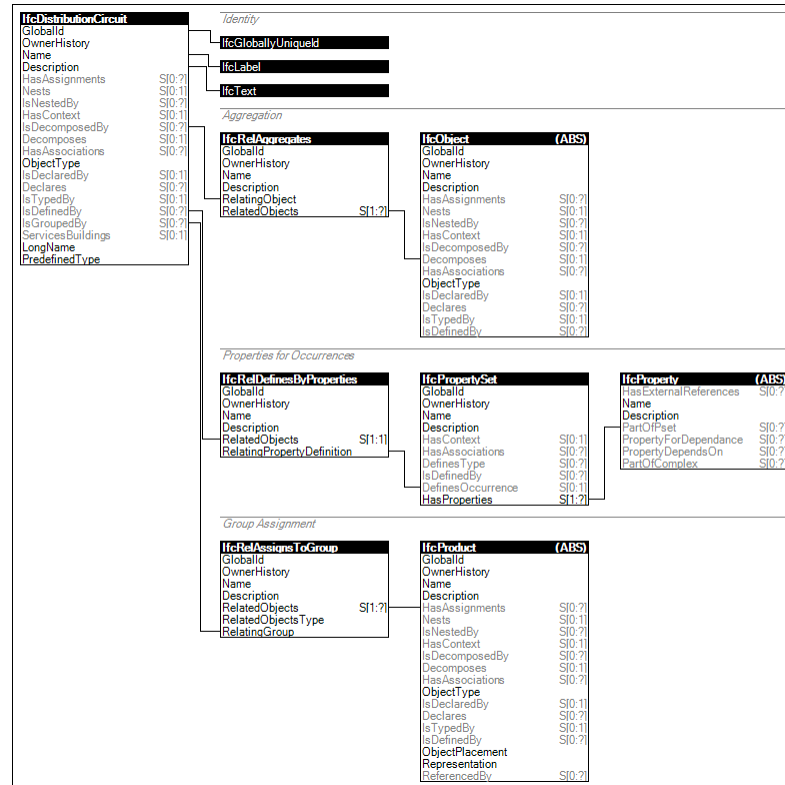
The surfacic geometric representation of *IfcWallElementedCase* is defined using the 'Surface' representation.

- *RepresentationIdentifier* : 'Surface'
- *RepresentationType* : 'Surface3D'

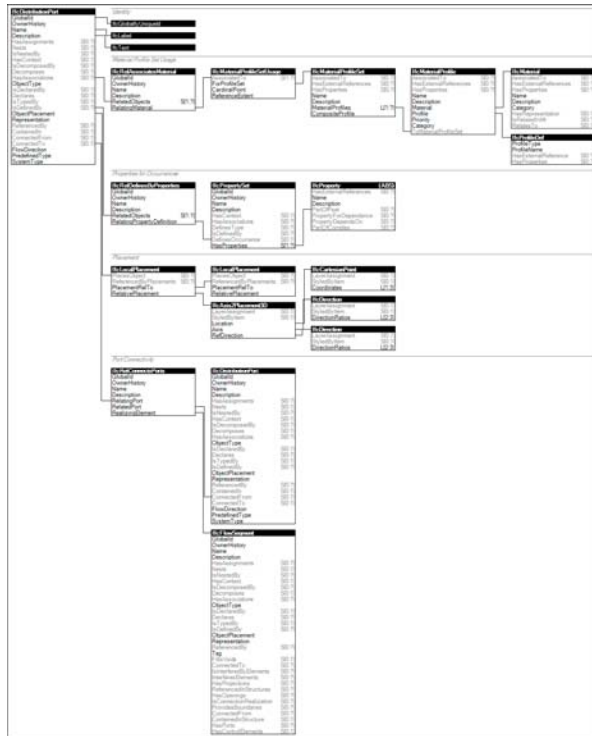
NOTE The 'Surface' can be used to define a surfacic model of the building (e.g. for analytical purposes, or for reduced Level of Detail representation). It could suppress the geometric details of the parts in the decomposition.

NOTE It is invalid to exchange a 'Body' shape representation of an *IfcWallElementedCase*. The body geometry is defined by the parts within the decomposition.

IfcDistributionCircuit



IfcDistributionPort



Port Connectivity

IfcDistributionPort may be connected to other objects as follows using the indicated relationship:

- **IfcDistributionPort** (*IfcRelConnectsPorts*) : Indicates a connection to another port having the same type and opposite flow direction. For port connections between elements, the *RelatingPort* is set to a port having *FlowDirection*=*SOURCE* and the *RelatedPort* is set to a port having *FlowDirection*=*SINK*. For aggregation scenarios, ports on a device may be mapped to aggregated devices within, in which case ports on the outer device indicate a single *FlowDirection* but have an additional connection internally to a port on an aggregated inner device. Refer to [IfcUnitaryEquipment](#) for an example.
- **IfcDistributionElement** (*IfcRelConnectsPortToElement*): For dynamic ports, indicates the containing element.

Figure 96 illustrates distribution port connectivity.

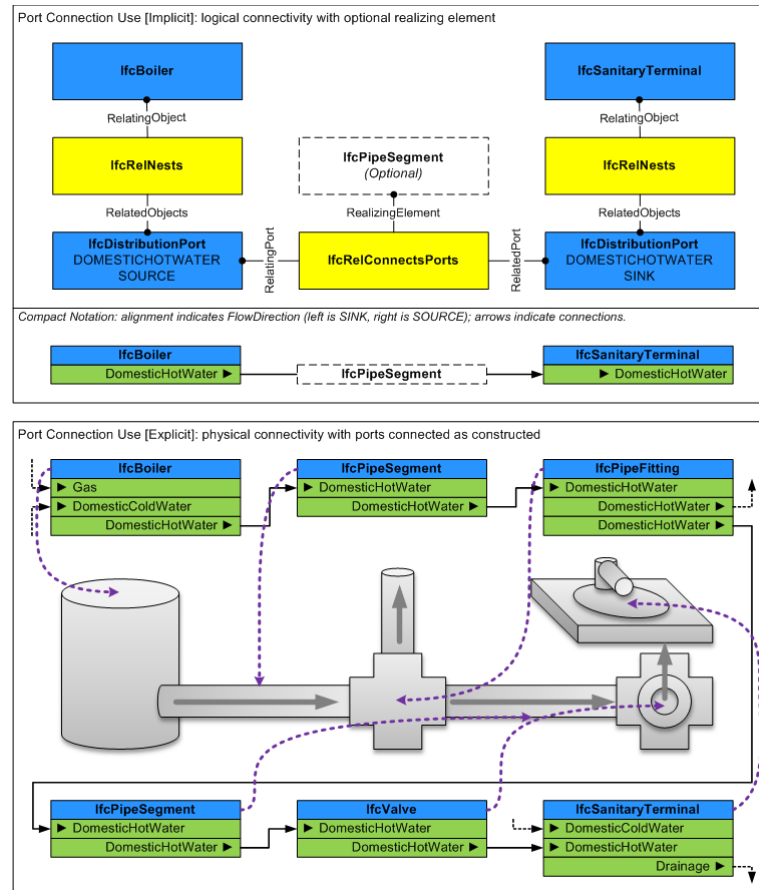


Figure 96 — Distribution port connectivity

Placement

The placement of a port indicates the position and orientation of how it may connect to a compatible port on another product. The placement shall be relative to the nesting *IfcDistributionElement*, *IfcDistributionElementType*, or enclosing *IfcDistributionPort*.

The *Location* is the midpoint of the physical connection, unless otherwise indicated by cardinal point on a material profile.

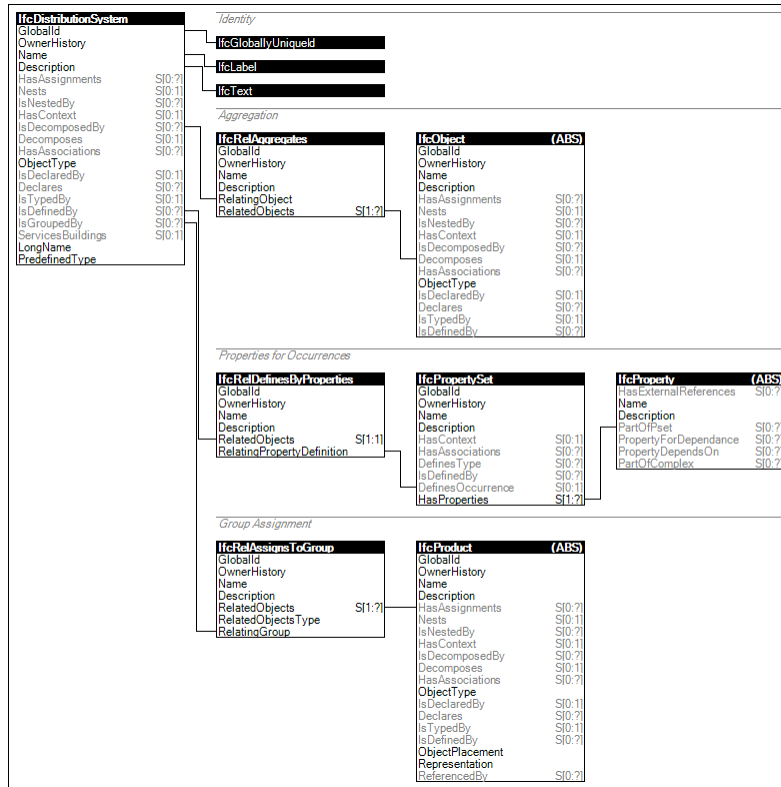
The *Axis* points in the direction of the physical connection away from the product if *FlowDirection* equals *SOURCE* (or *SOURCEANDSINK* or *NOTDEFINED*), or points opposite direction (to the product) if the *FlowDirection* equals *SINK*.

NOTE The rationale for positioning the *Axis* in the direction of flow is to allow for the same geometry to be used, such as for connectors with polarized cross-section.

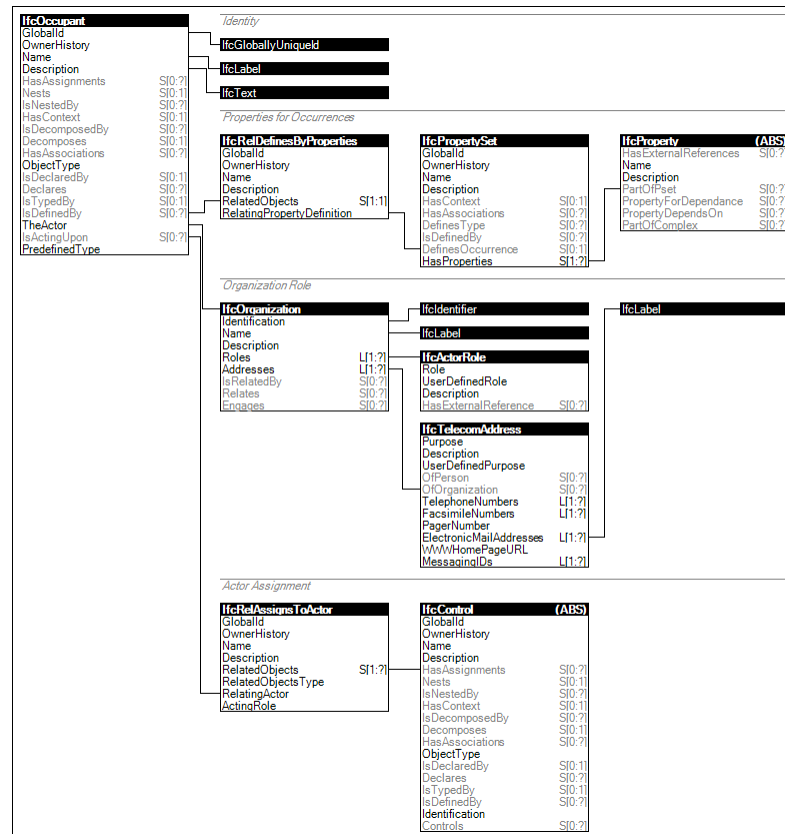
The *RefDirection* points in the direction of the local X axis of the material profile, where the local Y axis points up if looking towards the *Axis* where the local X axis points right.

Upon connecting elements through ports with rigid connections, each object shall be aligned such that the effective *Location*, *Axis*, and *RefDirection* of each port is aligned to be equal (with exception for circular profiles where the *RefDirection* need not be equal).

IfcDistributionSystem

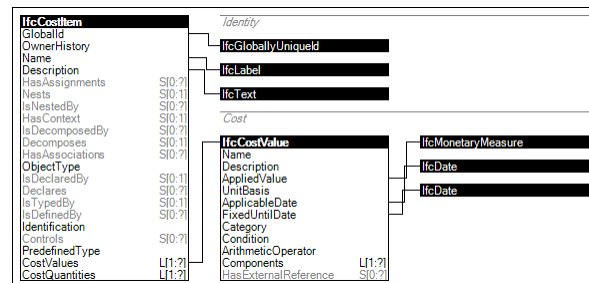


IfcOccupant

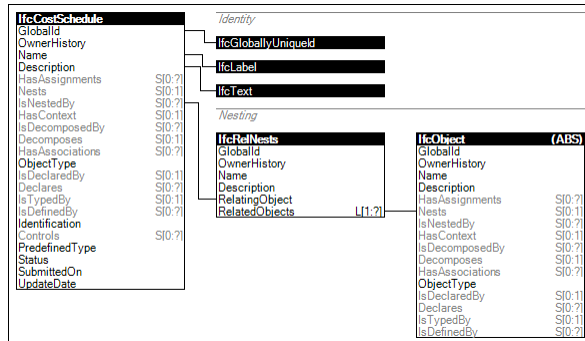


Actor Assignment

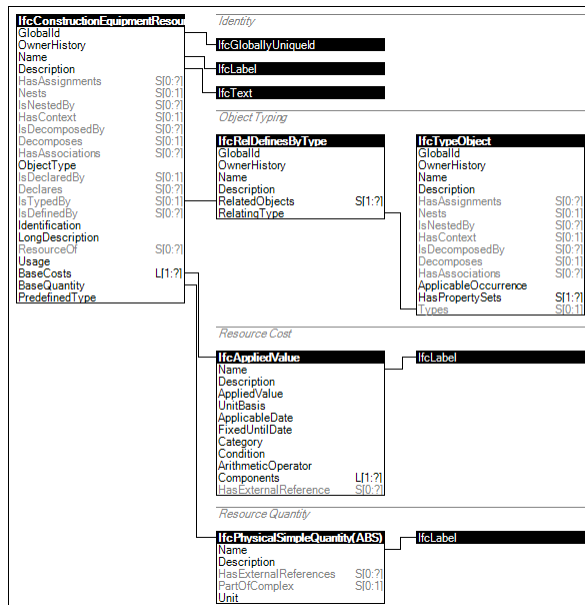
IfcCostItem



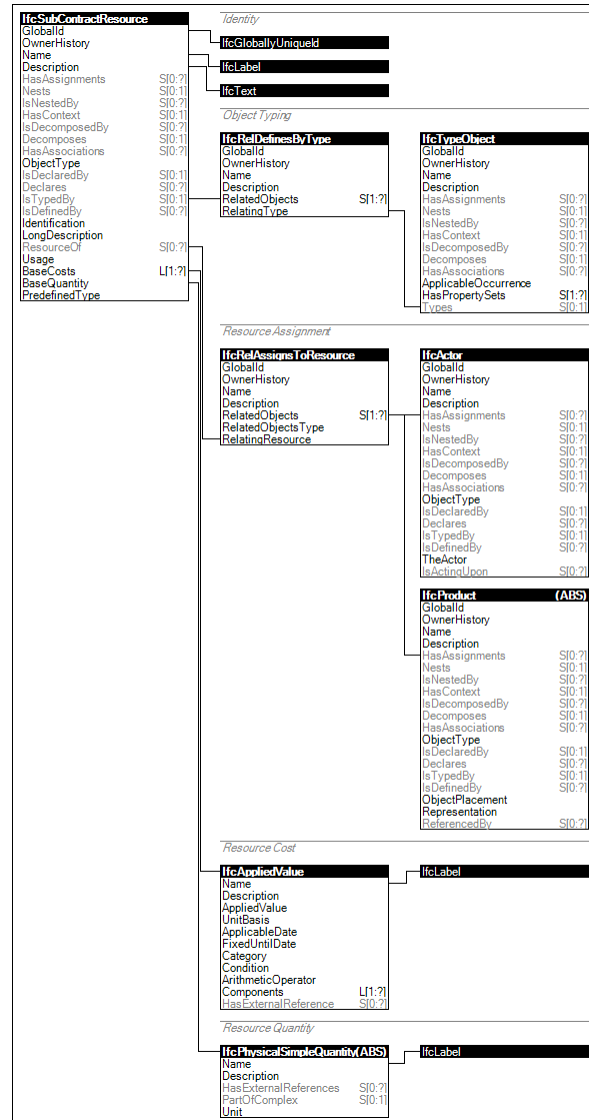
IfcCostSchedule



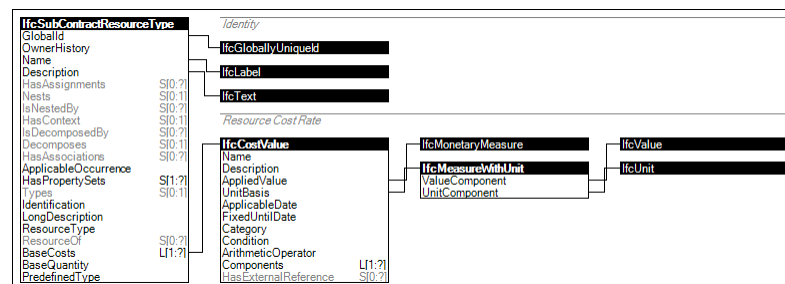
IfcConstructionEquipmentResource

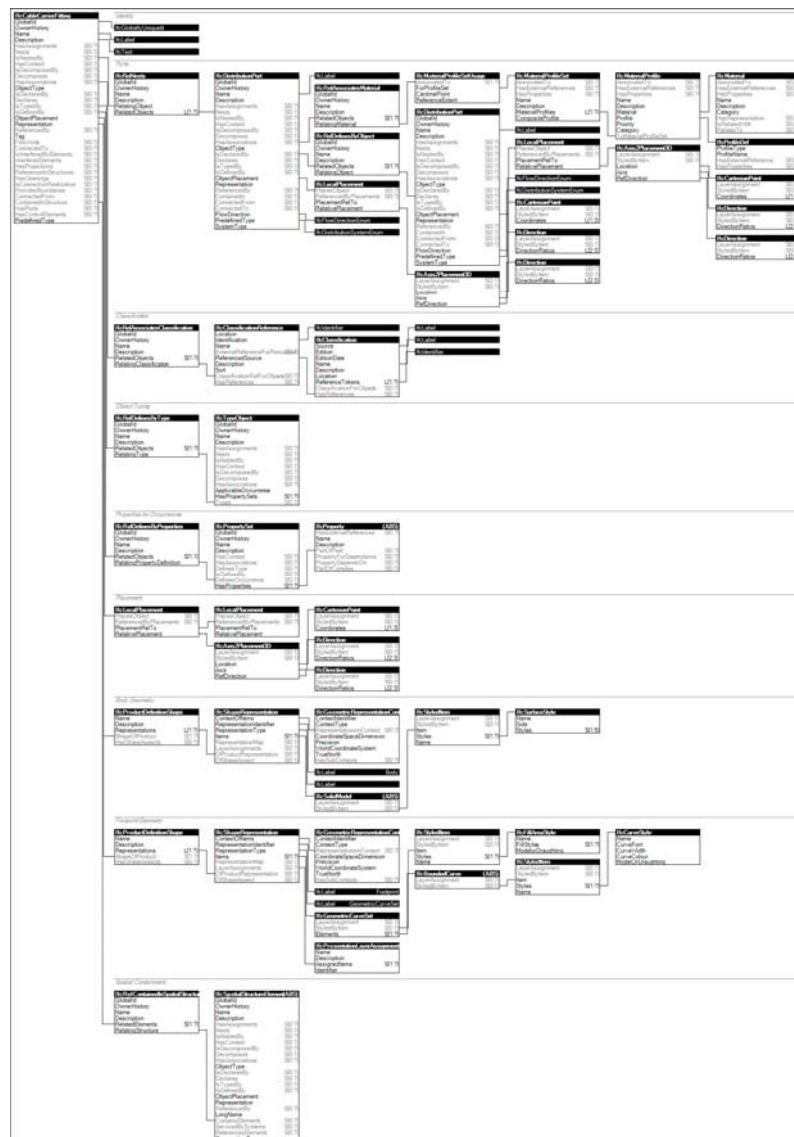


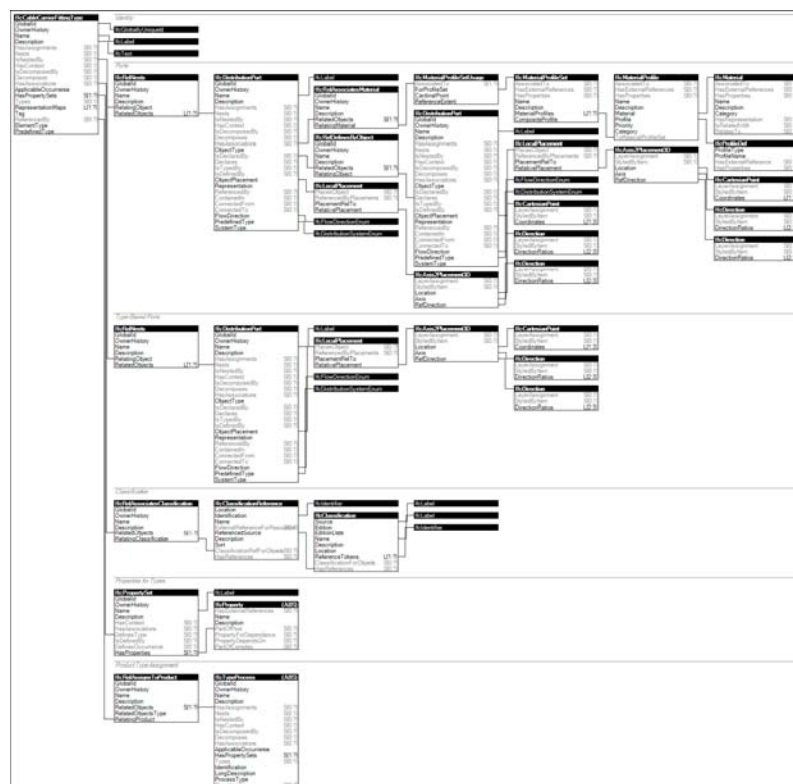
IfcSubContractResource



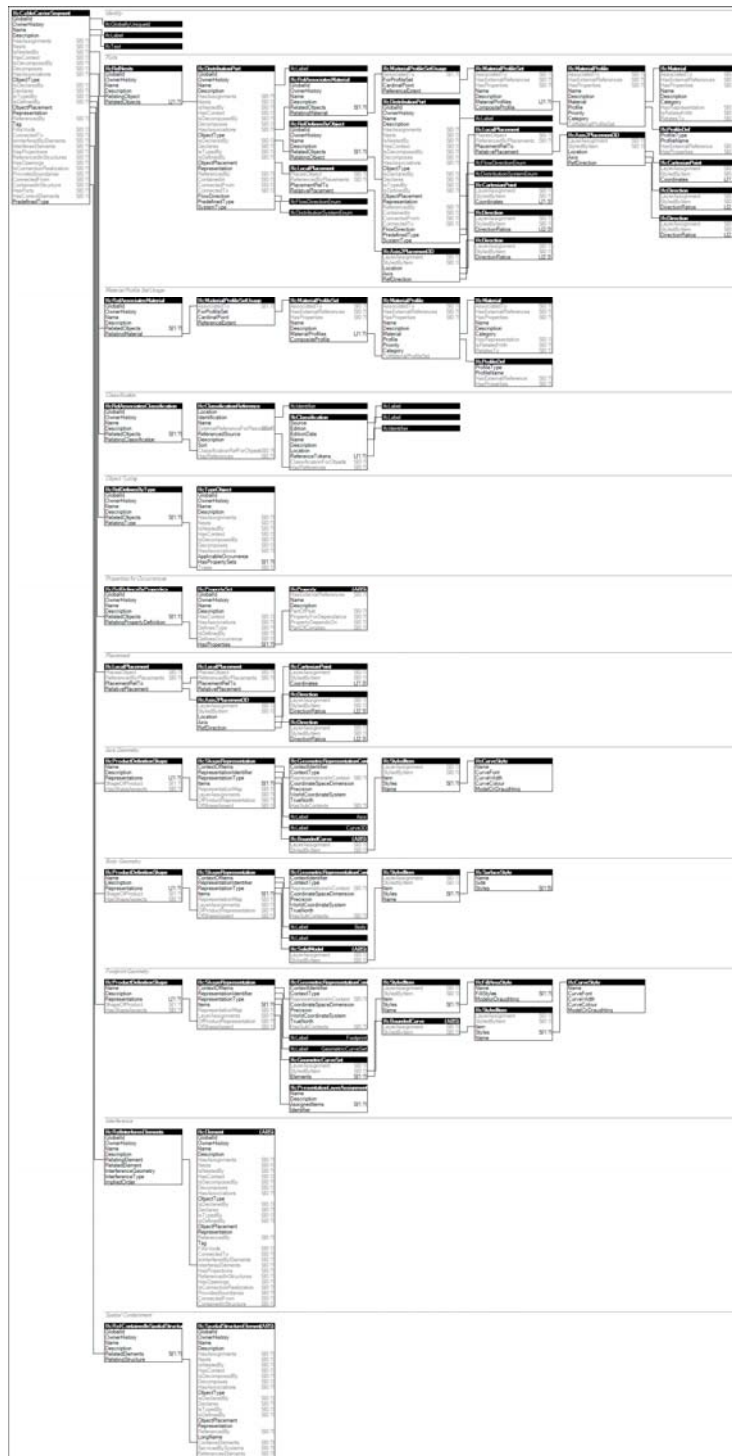
IfcSubContractResourceType



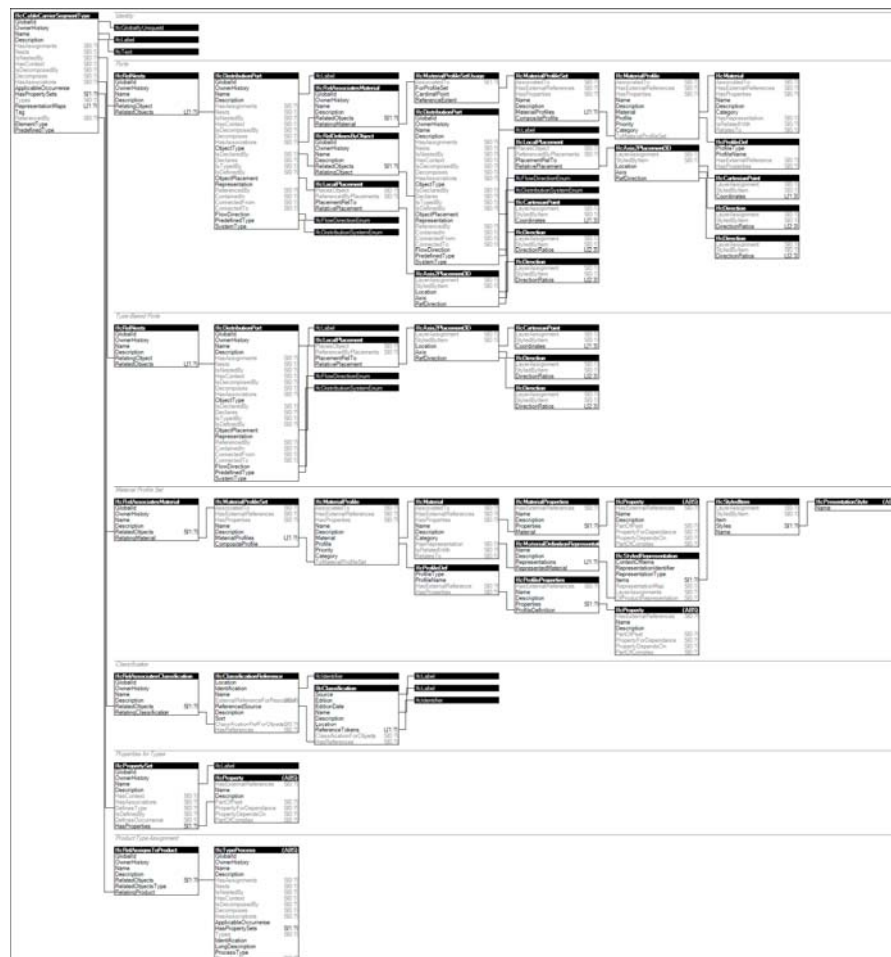




IfcCableCarrierSegment



IfcCableCarrierSegmentType

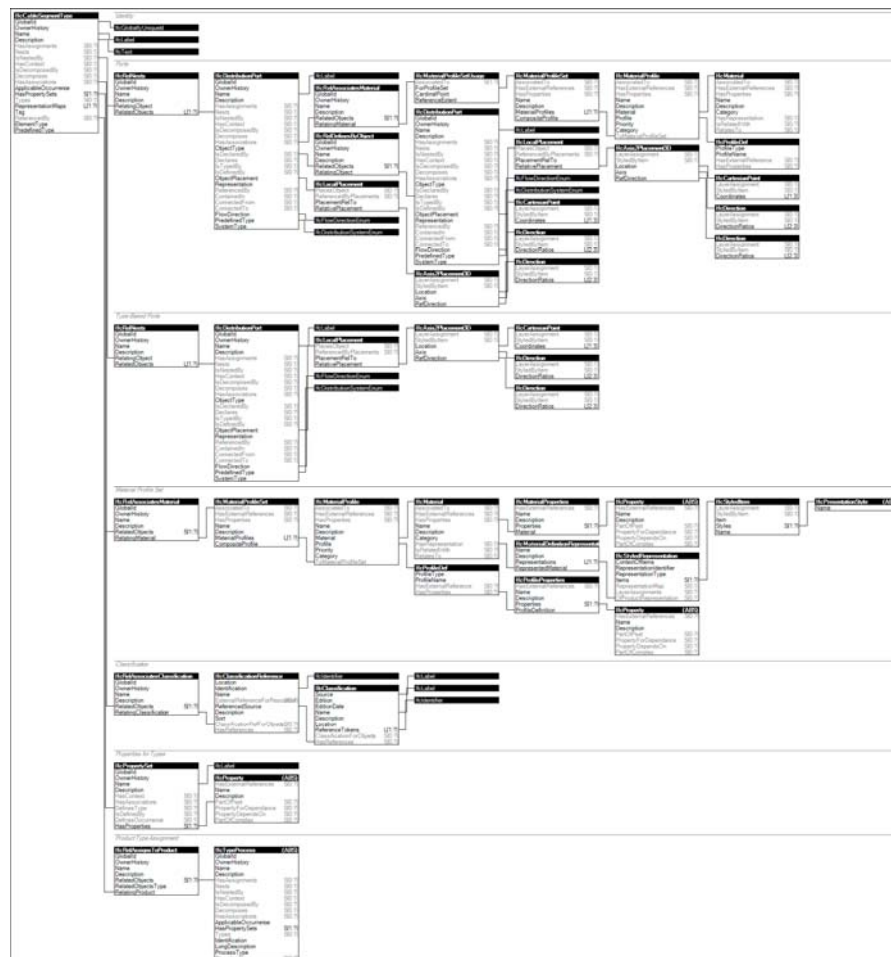


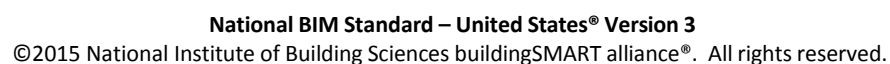


IfcCableSegment

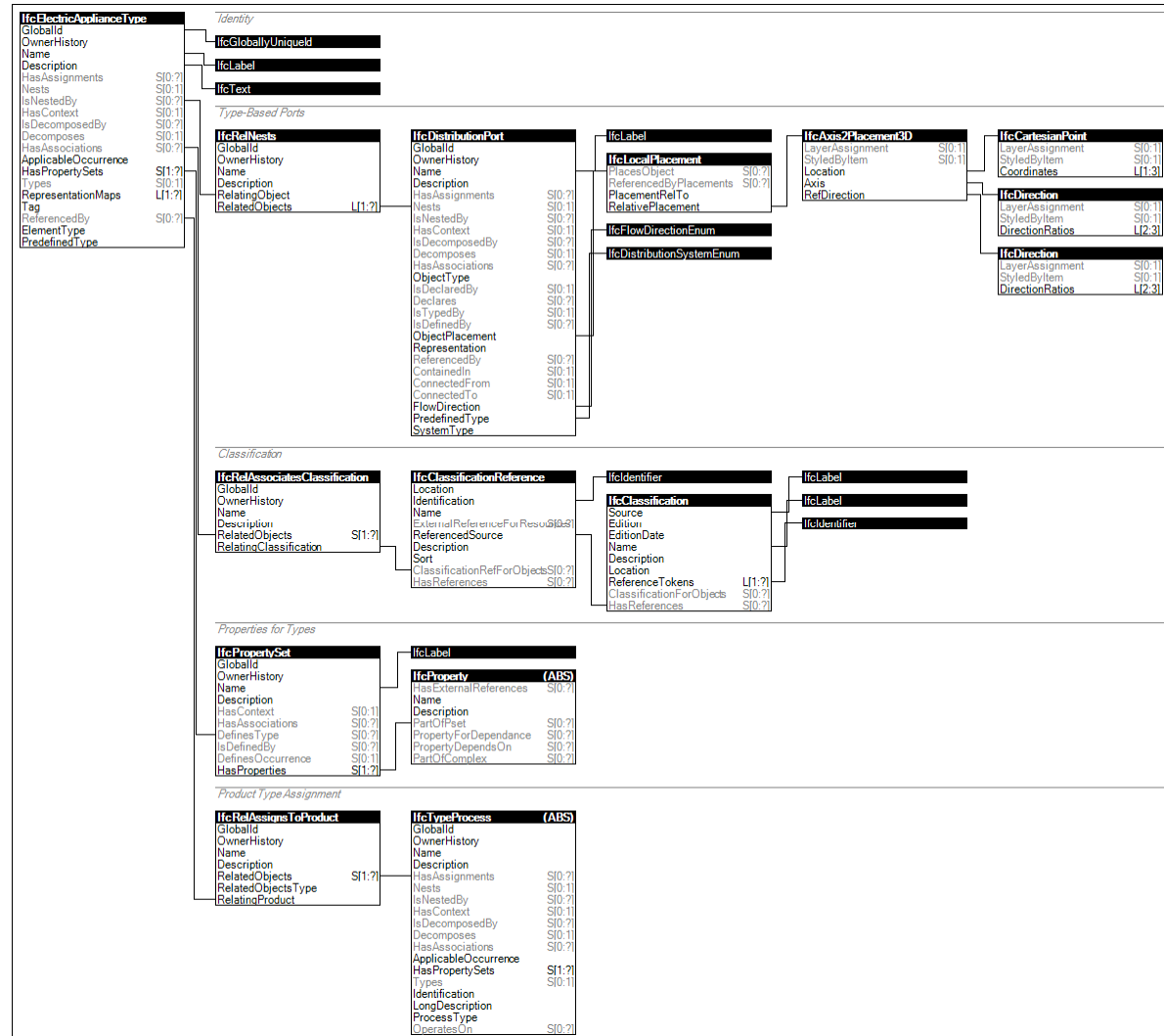


IfcCableSegmentType





IfcElectricApplianceType



IfcElectricDistributionBoard

As circuits on a distribution board are commonly numbered left-to-right then top-to-bottom, port ordering follows suit. A distribution board has a port corresponding to each slot that may be used by one or more breakers. For example, in a distribution board for a split-phase system having two columns, ports 1 and 2 are on the first phase, ports 3 and 4 are on the second phase, 5 and 6 on the first, and so on.

Single phase breakers (e.g. 120V) have one port connected to the slot of the distribution board, and one port optionally connected to a load. Double phase breakers (e.g. 240V) have two ports connected to slots within the distribution board, and one port optionally connected to a load, where such port encapsulates wires for both phases along with neutral and ground. Twin breakers have one port connected to the slot of the distribution board, and two ports optionally connected to separate loads.

Figure 109 illustrates multiple configurations of breakers within a distribution board.

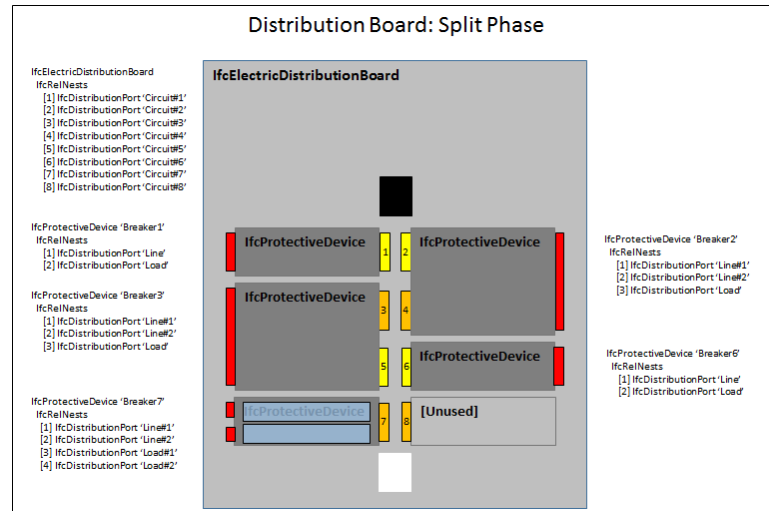


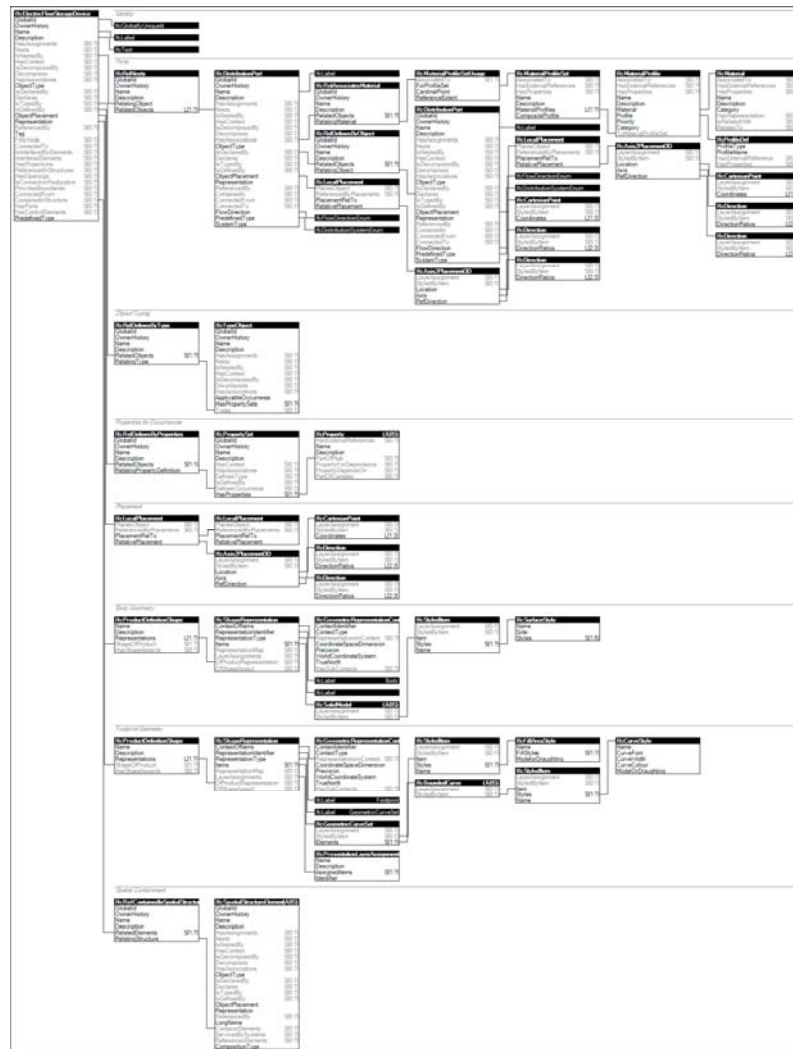
Figure 109 — Electric distribution board layout



IfcElectricDistributionBoardType

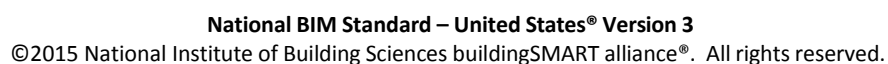
[illegible]

IfcElectricFlowStorageDevice



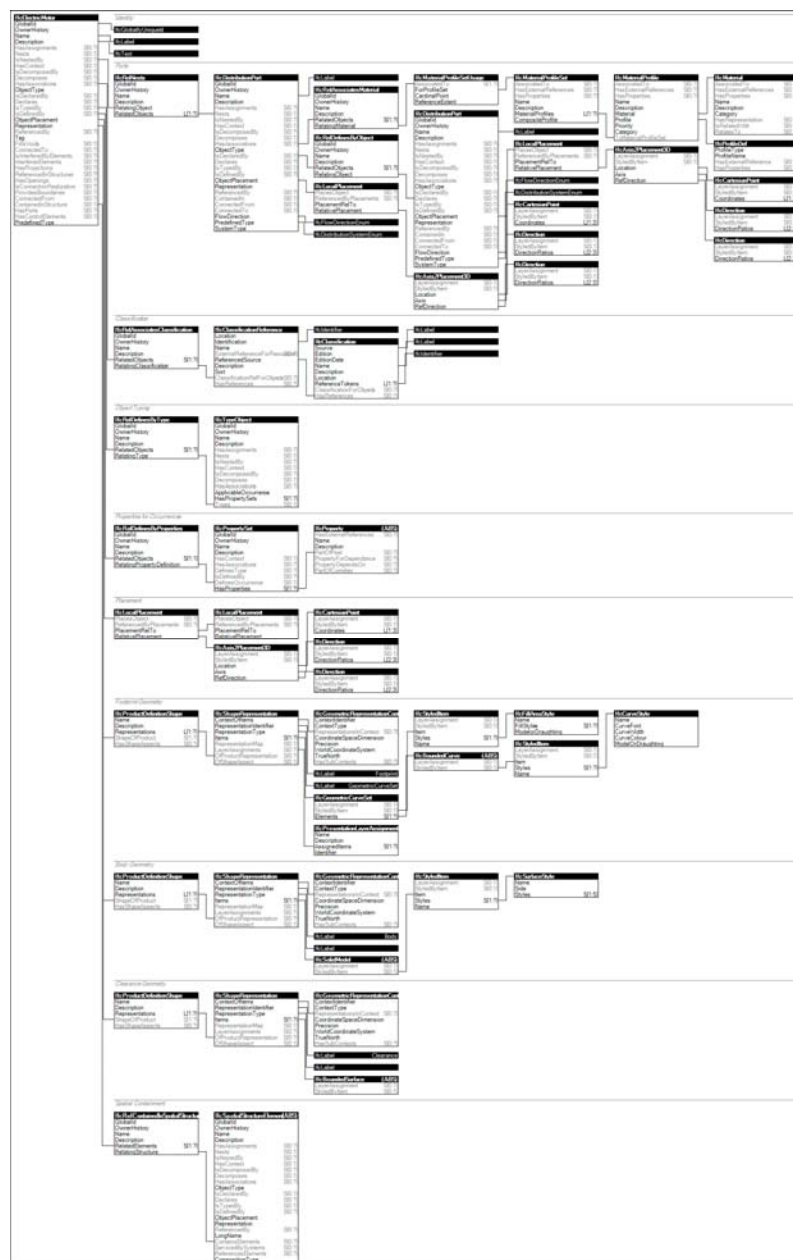
IfcElectricFlowStorageDeviceType

[illegible]

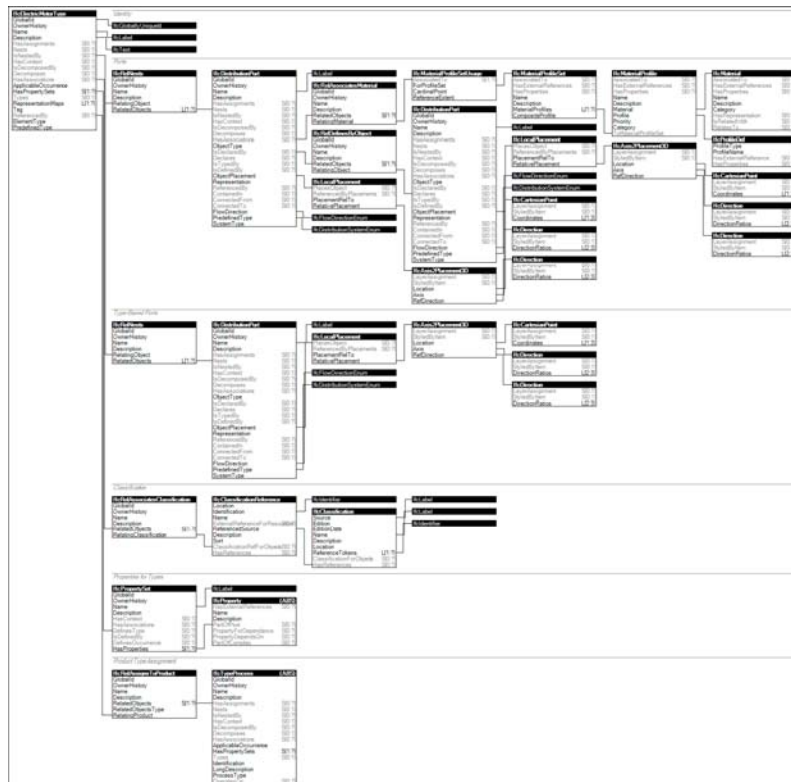


IfcElectricGeneratorType





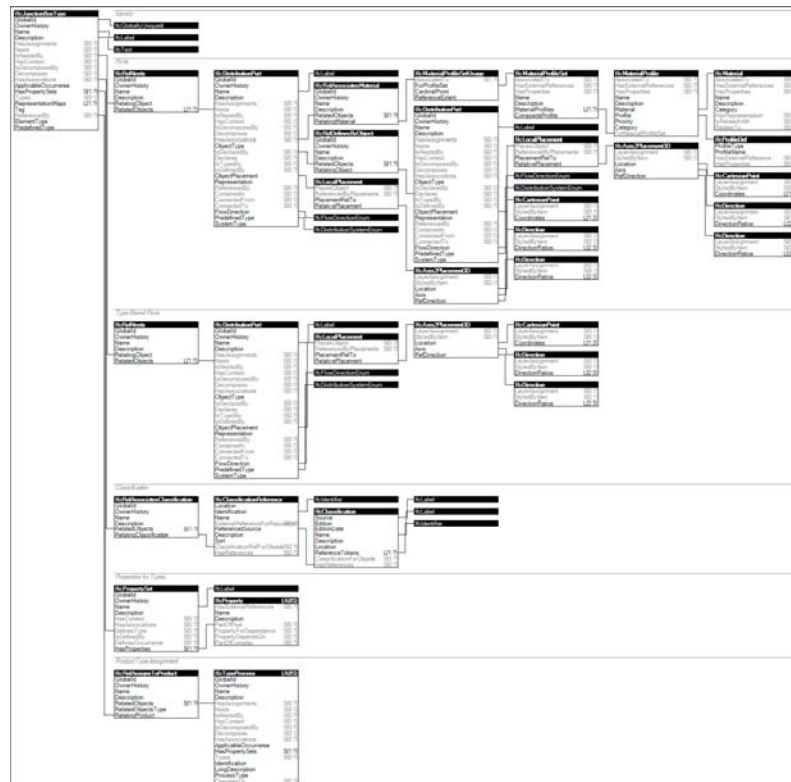
IfcElectricMotorType

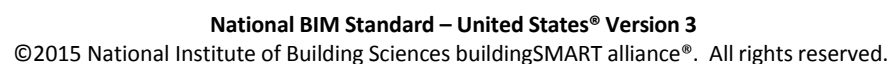


lfcJunctionBox

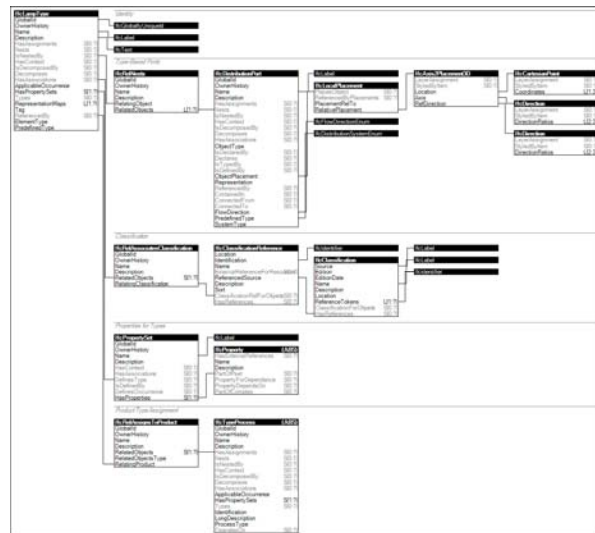


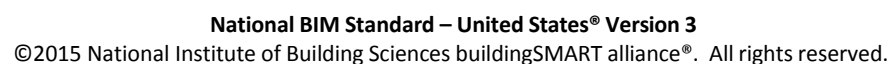
IfcJunctionBoxType





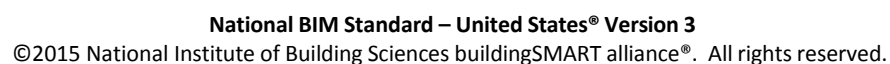
IfcLampType

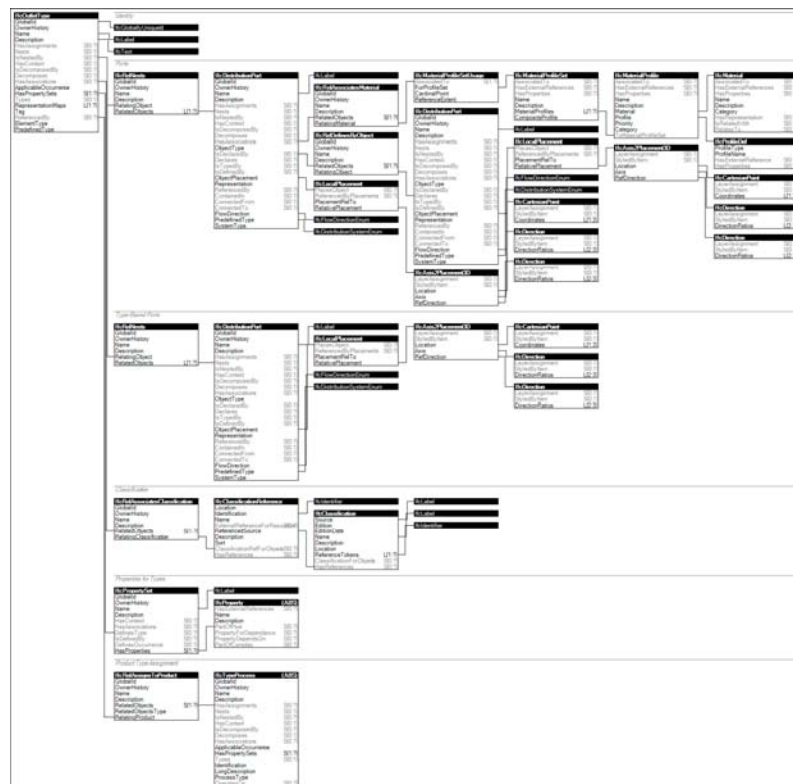




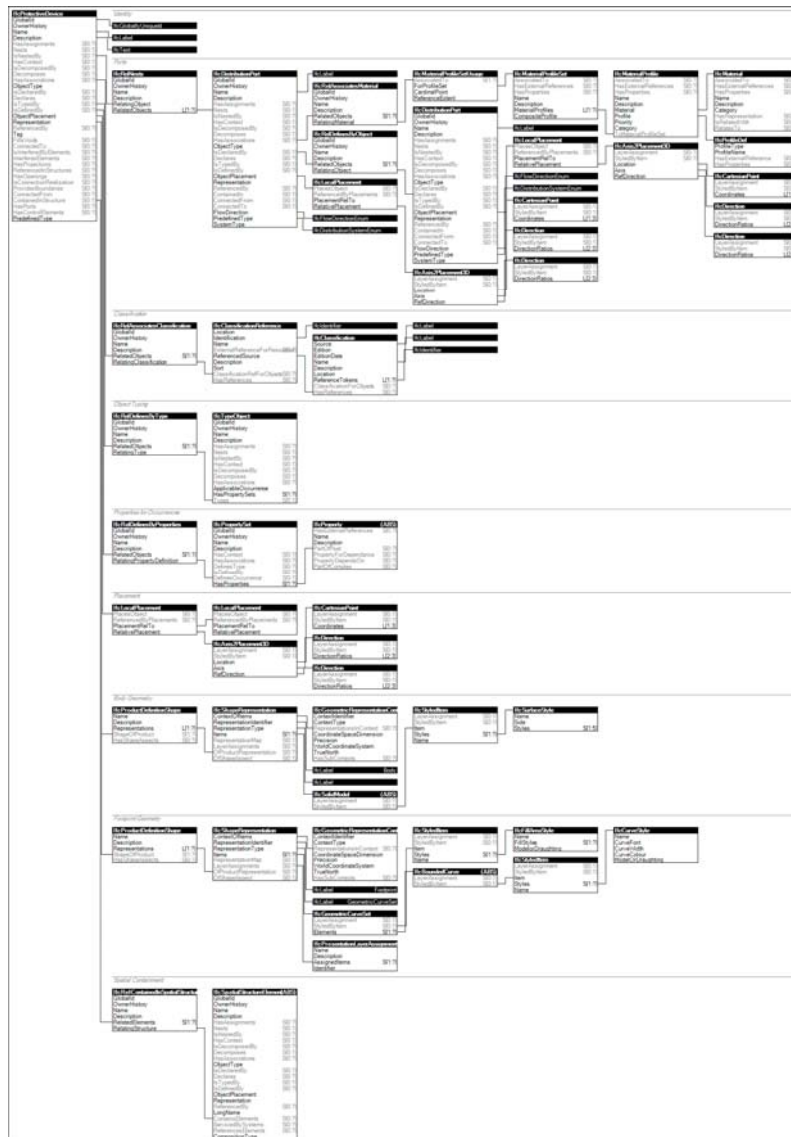
IfcLightFixtureType

[illegible]



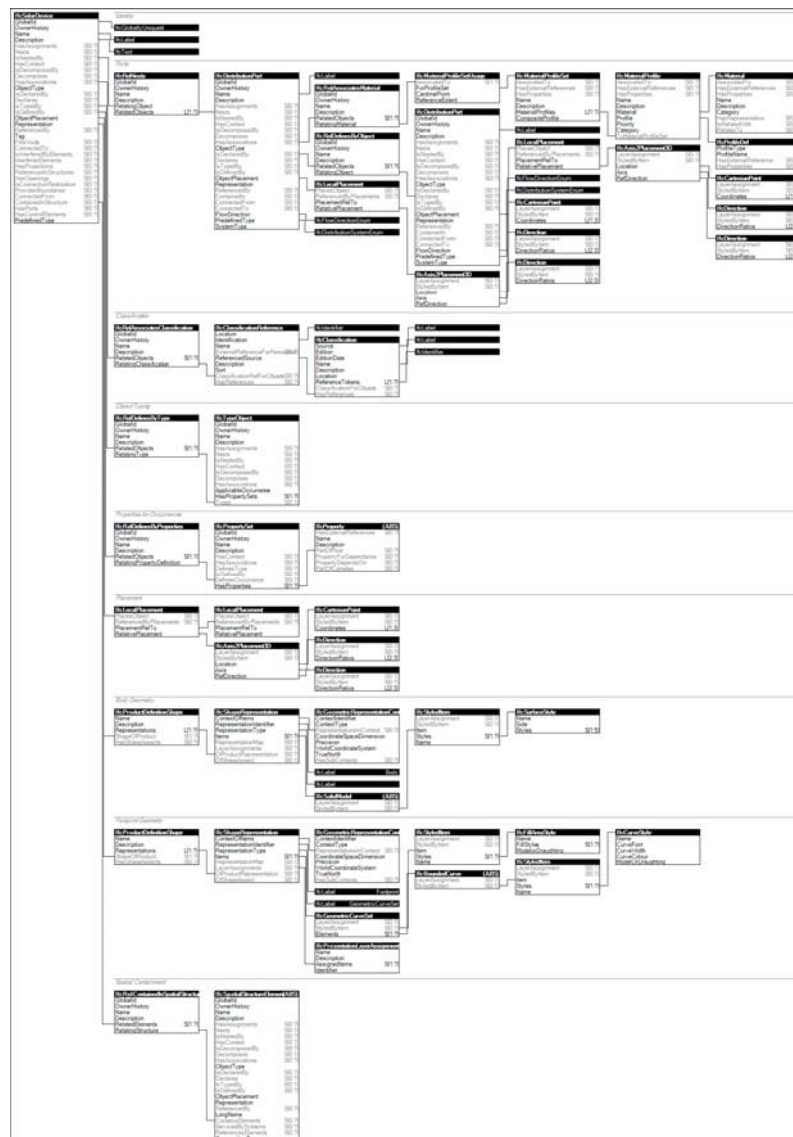


IfcProtectiveDevice

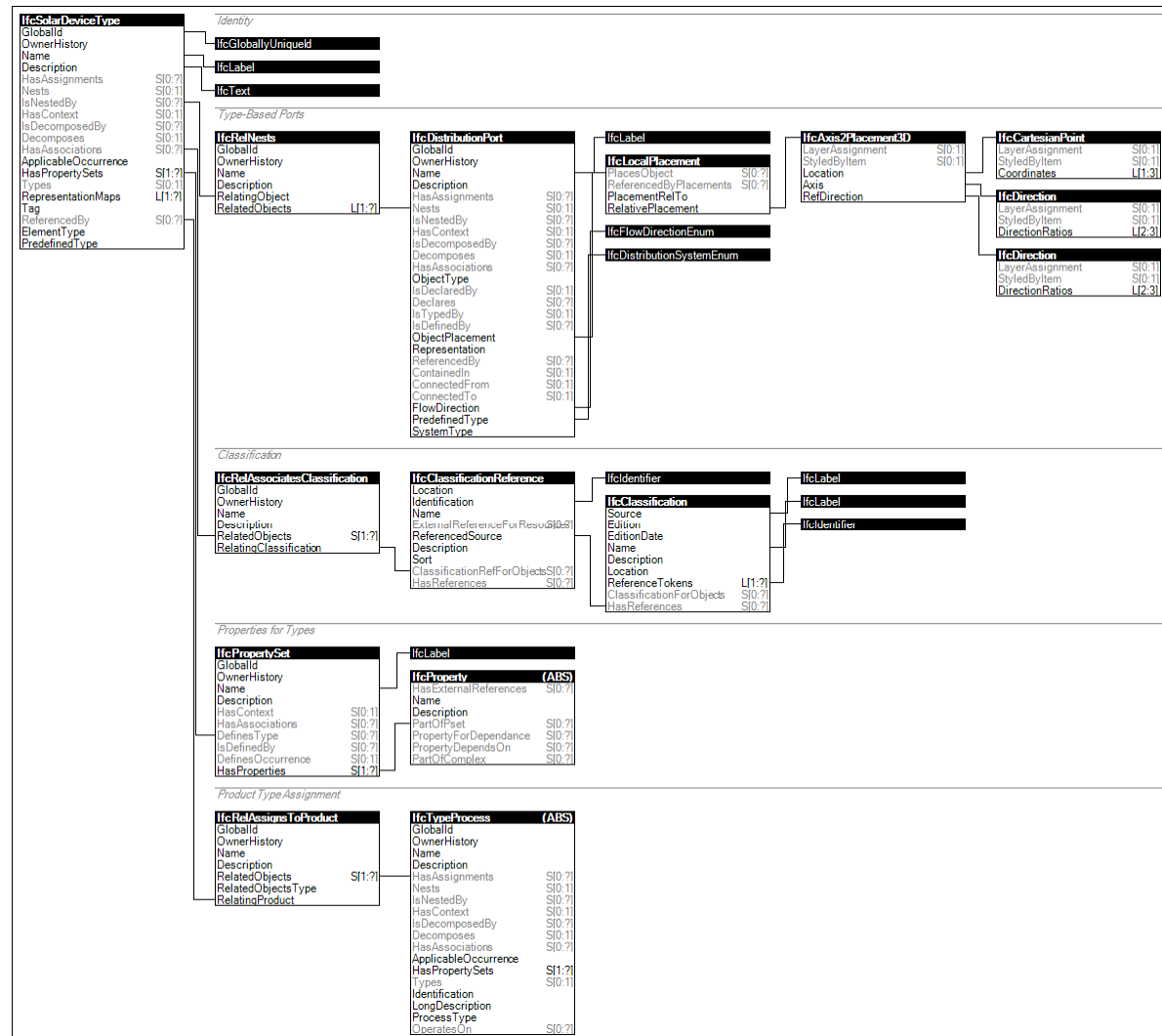


IfcProtectiveDeviceType





IfcSolarDeviceType



IfcSwitchingDevice

Switches may be defined in various configurations of 2-way, 3-way, or 4-way, with or without neutral wires, and with or without control capability (such as radio or powerline communication). Ports for switches reflect the logical flow of power and do not correspond to individual wires; this enables one-to-one connectivity between devices to reflect the design intent without requiring detailing of wire connections.

For the most typical configurations, switches are wired from the distribution panel where all conductors are available (Hot, Neutral, Ground) such as using 12/2-G wiring for a 120V circuit at 20A. Figure 110 illustrates a single 2-way switch controlling a light fixture. Figure 111 illustrates two 3-way switches controlling the same light fixture. Figure 112 illustrates three switches controlling the same light fixture, including two 3-way switches and one 4-way switch.

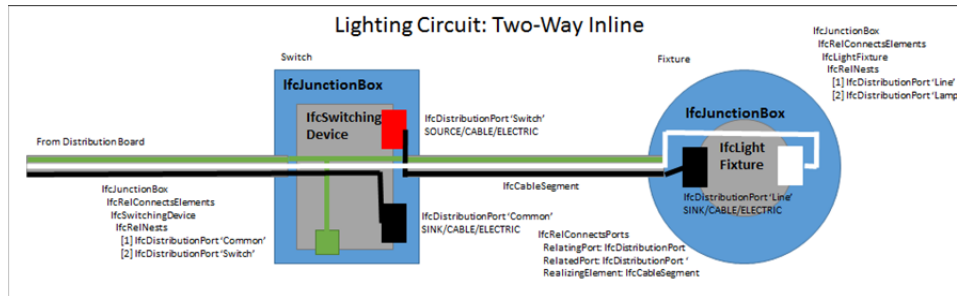


Figure 110 — Switching device layout for single switch

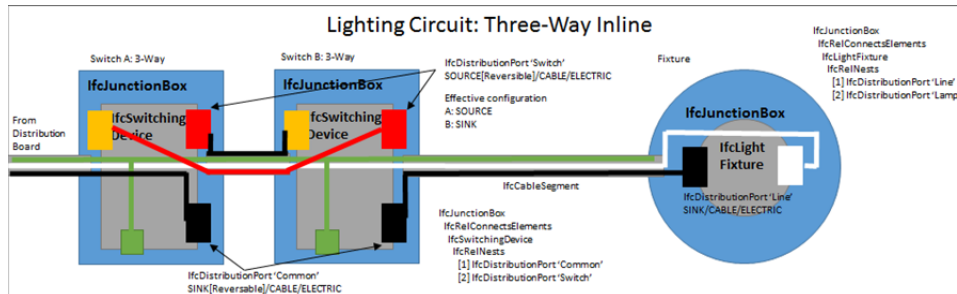


Figure 111 — Switching device layout for two switches

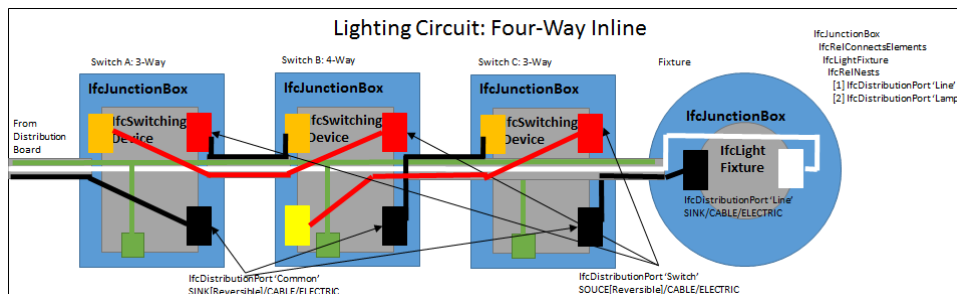


Figure 112 — Switching device layout for three switches

For some older configurations, switches may be wired as legs from the controlled light fixture, where one wire is hot and the other is switched hot such as using 14/2 wiring where no ground or neutral wire is available. For such configurations, the wiring path is indicated by using an *IfcCableFitting* within the *IfcJunctionBox* hosting the *IfcLightFixture*. Figure 113 illustrates such configuration for a single switch; multiple 3-way or 4-way switches may also use similar configuration.

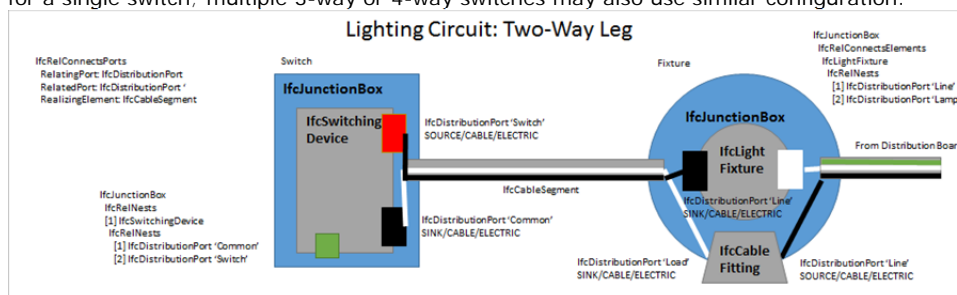


Figure 113 — Switching device layout for single switch leg

For switches that participate within a control system, an extra port is defined for sending and receiving messages within the control system. Such switches may not necessarily have a load attached if they serve as a remote switch for another load-bearing switch. As such switches draw power themselves for illumination and communication, most require either a neutral wire or else an attached load capable of running at lower power. Switches that control multiple loads are also possible, such as keypads with 8 buttons; while there may be virtual connections between such buttons and other switches, only connections impacting physical wiring are indicated by ports. Figure 114 illustrates a single controlled switch. Figure 115 indicates two controlled switches. Figure 116 indicates three controlled switches. Figure 117 illustrates a single controlled switch without neutral wire.

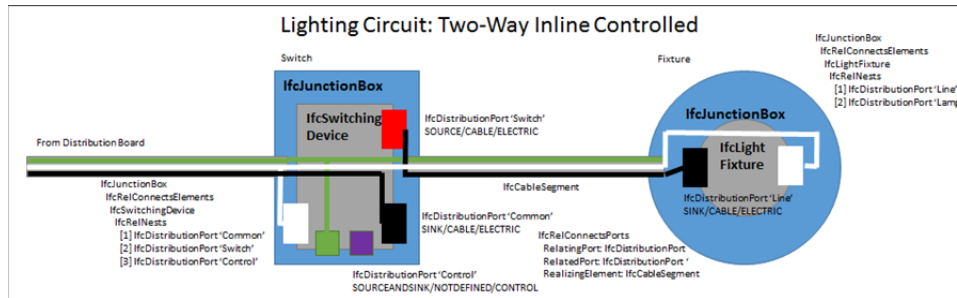


Figure 114 — Switching device layout for single switch controlled

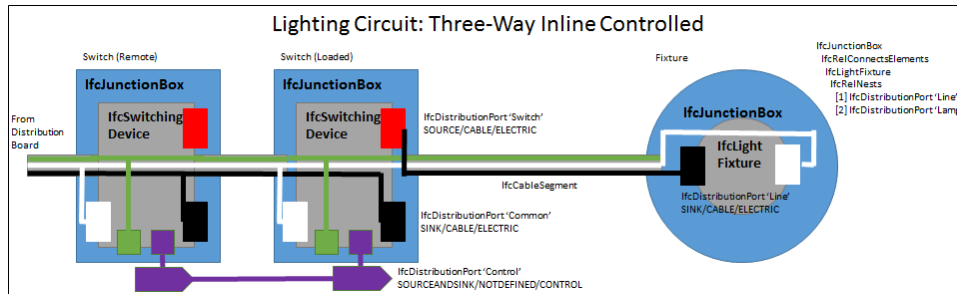


Figure 115 — Switching device layout for two switches controlled

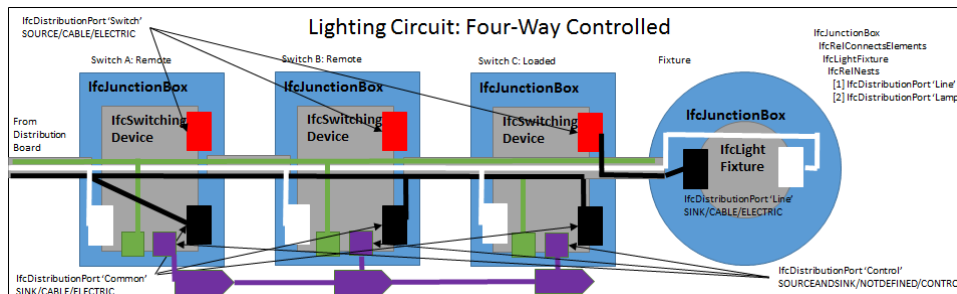


Figure 116 — Switching device layout for three switches controlled

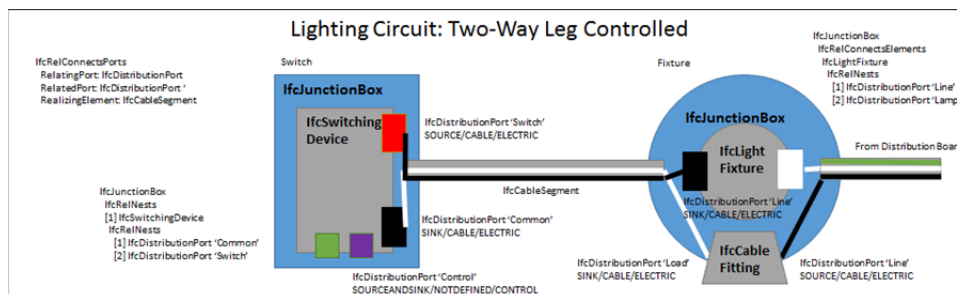


Figure 117 — Switching device layout for single switch leg controlled

IfcSwitchingDeviceType



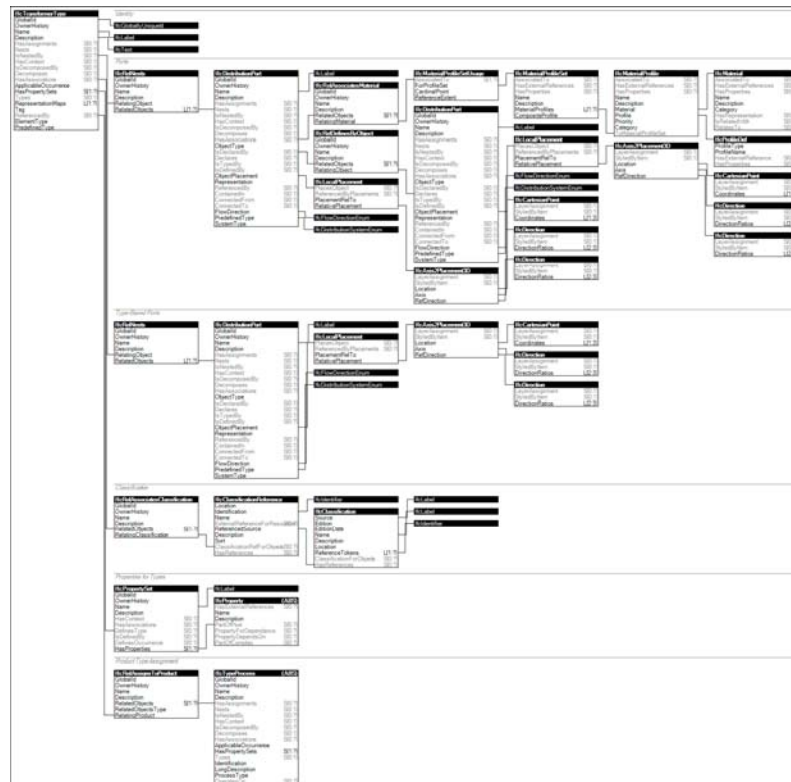
IfcTransformer

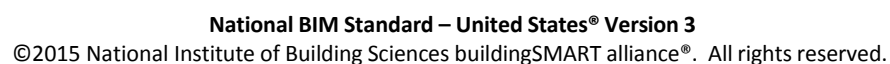


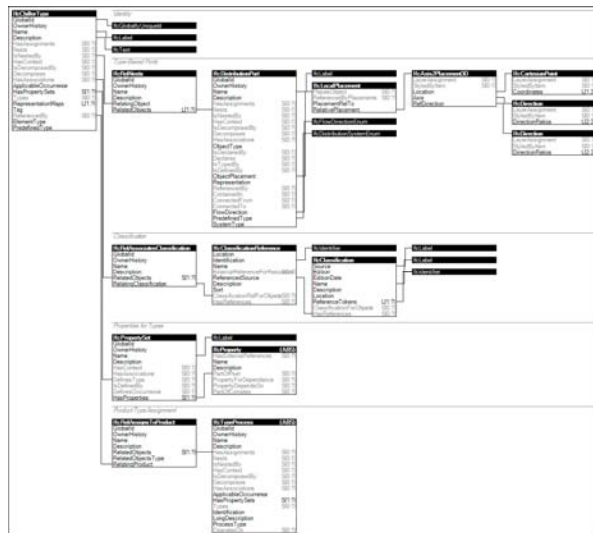
Spatial Containment

Transformers for low voltage lighting (24V) require space to be allocated above ceilings or in equipment rooms that may require coordination with other trades.

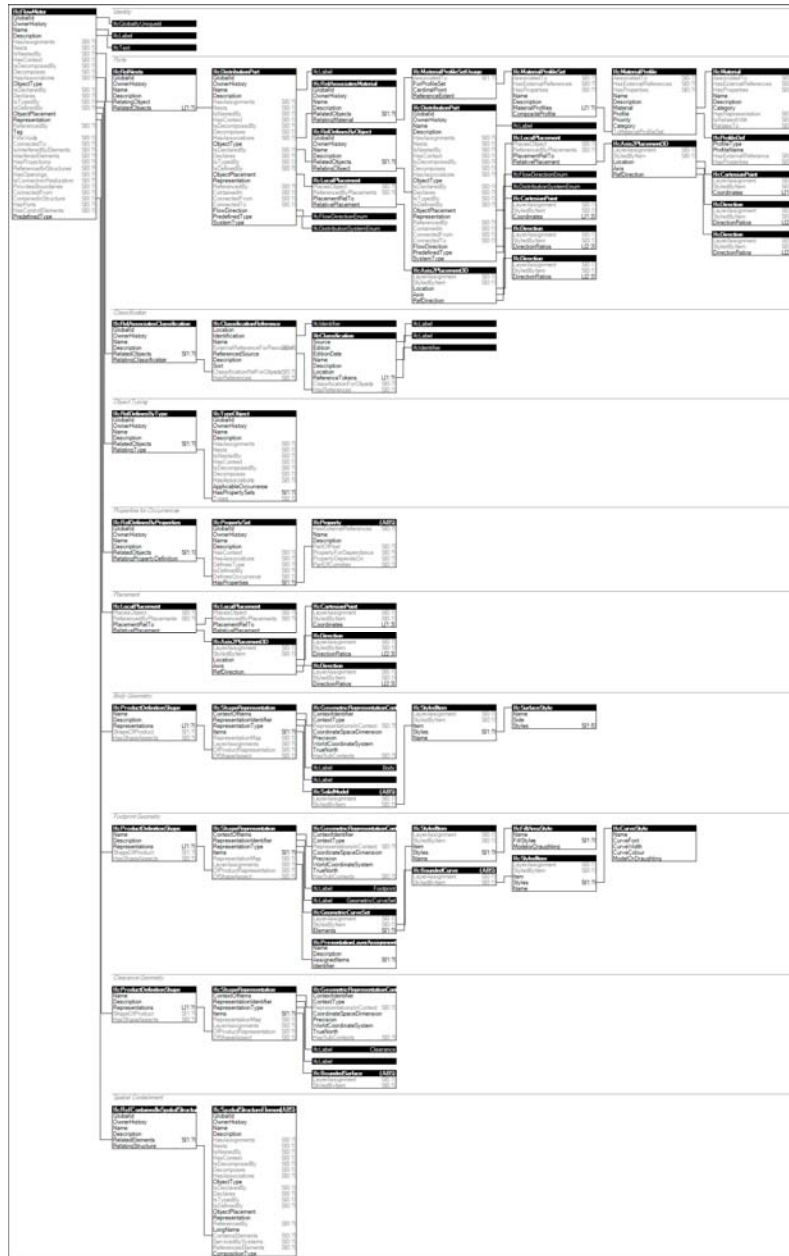
lfcTransformerType

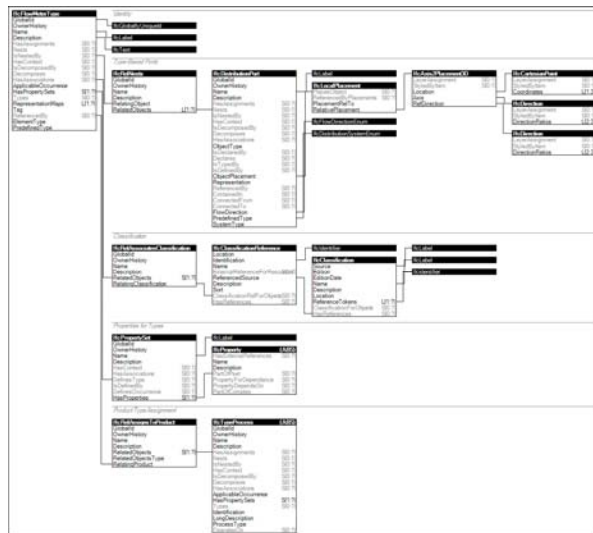


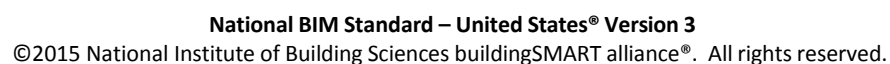


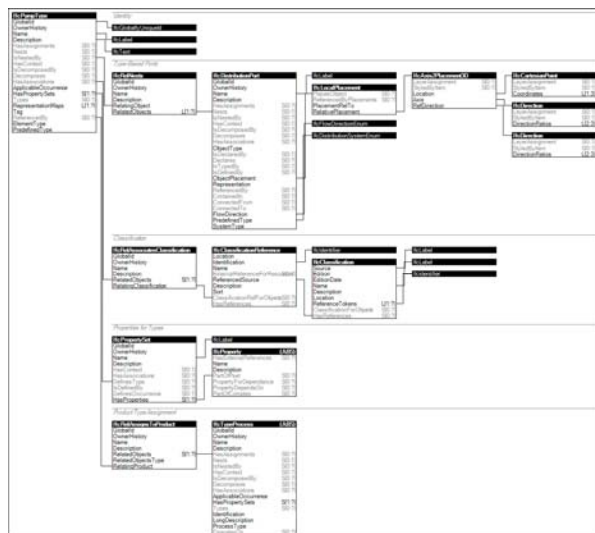


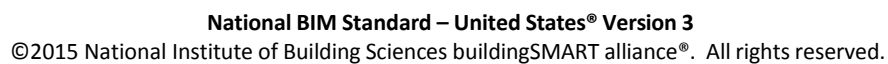
IfcFlowMeter



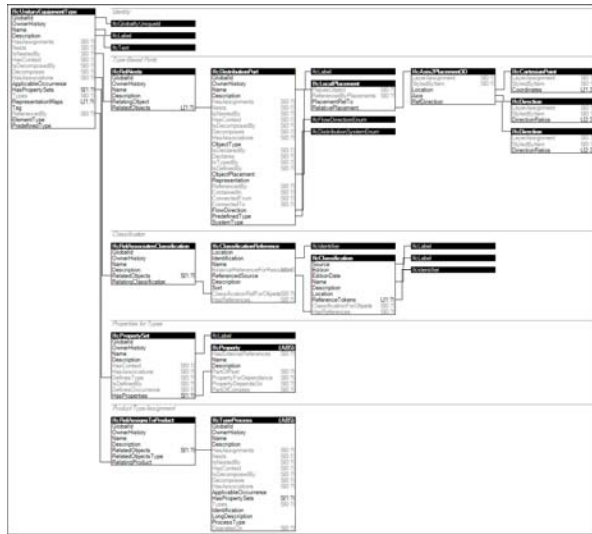








IfcUnitaryEquipmentType



4.7.6.2.3 Concept attributes list

Concepts may be defined that use parameters to indicate applicable values. For example, plumbing objects may make use of ports to enable connectivity to other objects for distribution of water, and a specific entity such as a hot water heater may have specific ports such as “ColdWaterIn” and “HotWaterOut”. Defining attributes at concepts enables re-use of concepts where the data structures are the same, but applicable values may differ. Each concept is shown in a subsection as follows, with rows corresponding to entities and rule instances, columns corresponding to template parameters, and cells corresponding to values applied to rules.

Properties for Occurrences

Entity	PredefinedType	Name
IfcActor		Pset_ActorCommon
IfcBuilding		Pset_BuildingCommon
		Pset_UtilityConsumptionPHistory
IfcBuildingStorey		Pset_BuildingStoreyCommon
IfcDistributionElement		Pset_ElectricalDeviceCommon
IfcSpace		SPARKie_SpaceElectricalRequirements
		Pset_SpaceLightingRequirements
		Pset_SpaceOccupancyRequirements
		Pset_SpaceCommon
IfcTransportElement		
IfcDistributionCircuit		
IfcDistributionPort	CABLE	Pset_DistributionPortTypeCable
IfcDistributionSystem		Pset_DistributionSystemCommon
	ELECTRICAL	Pset_DistributionSystemTypeElectrical
	ELECTRICAL	SPARKie_DistributionSystemTypeElectrical

Entity	PredefinedType	Name
IfcOccupant		
IfcCableCarrierFitting		Pset_CableCarrierFittingTypeCommon
IfcCableCarrierSegment	CABLELADDERSEGMENT	Pset_CableCarrierSegmentTypeCableLadderSegment
	CABLETRAYSEGMENT	Pset_CableCarrierSegmentTypeCableTraySegment
	CABLETRUNKINGSEGMENT	Pset_CableCarrierSegmentTypeCableTrunkingSegment
		Pset_CableCarrierSegmentTypeCommon
	CONDUITSEGMENT	Pset_CableCarrierSegmentTypeConduitSegment
IfcCableFitting		Pset_CableFittingTypeCommon
IfcCableSegment	BUSBARSEGMENT	Pset_CableSegmentTypeBusBarSegment
	CABLESEGMENT	Pset_CableSegmentTypeCableSegment
		Pset_CableSegmentTypeCommon
	CONDUCTORSEGMENT	Pset_CableSegmentTypeConductorSegment
	CORESEGMENT	Pset_CableSegmentTypeCoreSegment
		Pset_CableSegmentOccurrence
IfcElectricAppliance		Pset_ElectricAppliancePHistory
		Pset_ElectricApplianceTypeCommon
	DISHWASHER	Pset_ElectricApplianceTypeDishwasher
	ELECTRICCOOKER	Pset_ElectricApplianceTypeElectricCooker
IfcElectricDistributionBoard		Pset_ElectricDistributionBoardTypeCommon
		Pset_ElectricDistributionBoardOccurrence
IfcElectricFlowStorageDevice		Pset_ElectricFlowStorageDeviceTypeCommon
IfcElectricGenerator		Pset_ElectricGeneratorTypeCommon
IfcElectricMotor		
IfcJunctionBox		Pset_JunctionBoxTypeCommon
IfcLamp		
IfcLightFixture		Pset_LightFixtureTypeCommon
	SECURITYLIGHTING	Pset_LightFixtureTypeSecurityLighting
IfcOutlet		Pset_OutletTypeCommon
IfcProtectiveDevice		Pset_ProtectiveDeviceBreakerUnitI2TCurve
		Pset_ProtectiveDeviceBreakerUnitI2TFuseCurve
		Pset_ProtectiveDeviceBreakerUnitIPICurve
	CIRCUITBREAKER	Pset_ProtectiveDeviceBreakerUnitTypeMCB
		Pset_ProtectiveDeviceBreakerUnitTypeMotorProtection
		Pset_ProtectiveDeviceTrippingCurve
	CIRCUITBREAKER	Pset_ProtectiveDeviceTypeCircuitBreaker
		Pset_ProtectiveDeviceTypeCommon
	EARTHLEAKAGECIRCUITBREAKER	Pset_ProtectiveDeviceTypeEarthLeakageCircuitBreaker
	FUSEDISCONNECTOR	Pset_ProtectiveDeviceTypeFuseDisconnecter

Entity	PredefinedType	Name
	RESIDUALCURRENTCIRCUITBREAKER	Pset_ProtectiveDeviceTypeResidualCurrentCircuitBreaker
	RESIDUALCURRENTSWITCH	Pset_ProtectiveDeviceTypeResidualCurrentSwitch
	VARISTOR	Pset_ProtectiveDeviceTypeVaristor
IfcSolarDevice		Pset_SolarDeviceTypeCommon
IfcSwitchingDevice		
IfcTransformer		
IfcChiller		
IfcFlowMeter		
		Pset_FlowMeterOccurrence
		Pset_FlowMeterTypeCommon
	ENERGYMETER	Pset_FlowMeterTypeEnergyMeter
IfcPump		
IfcUnitaryEquipment		

Organization Role

Entity	Role	UserDefinedRole
IfcActor	USERDEFINED	ElectricalUtility
IfcOccupant		

Actor Assignment

Entity	Type
IfcActor	IfcWorkPlan
IfcOccupant	IfcSpatialStructureElement
	IfcWorkCalendar

Spatial Decomposition

Entity	Spatial Parts
IfcProject	IfcSite
IfcBuilding	IfcBuildingStorey
IfcBuildingStorey	
IfcSite	IfcBuilding

Project Context

Entity	ContextIdentifier	ContextType
IfcProject		

Project Declaration

Entity	Type
IfcProject	IfcSpaceType
	IfcOccupant

	IfcActor
	IfcSpace
	IfcWorkPlan
	IfcDistributionElementType
	IfcDistributionSystem

Control Assignment

Entity	Type
IfcPerformanceHistory	IfcGroup
	IfcProduct
	IfcProcess
	IfcResource
IfcWorkCalendar	IfcSpatialStructureElement
IfcWorkPlan	IfcDistributionSystem

Process Assignment

Entity	Type
IfcTask	IfcSubContractResource

Nesting

Entity	Type
IfcTaskType	
IfcWorkCalendar	IfcCostSchedule
IfcCostSchedule	IfcCostItem

Process Type Assignment

Entity	Type
IfcTaskType	IfcSubContractResourceType

Aggregation

Entity	PredefinedType	RelatedObjects
IfcWorkPlan		IfcWorkCalendar
		IfcPerformanceHistory
IfcSlabElementedCase		IfcBeam
		IfcPlate
		IfcBuildingElementPart
IfcWallElementedCase		IfcMember
		IfcPlate
		IfcBuildingElementPart
		IfcBeam
IfcDistributionSystem		IfcDistributionCircuit

Classification

Entity	Source	Name	Tokens
IfcBuilding	CSI	OmniClass	11-00-00-00
IfcSpace	CSI	OmniClass	13-00 00 00
IfcSpaceType	CSI	OmniClass	13-00 00 00
IfcTransportElement			
IfcTransportElementType			
IfcCableCarrierFitting			
IfcCableCarrierFittingType			
IfcCableCarrierSegment			
IfcCableCarrierSegmentType			
IfcCableFitting			
IfcCableSegment			
IfcCableSegmentType			
IfcElectricAppliance			
IfcElectricApplianceType			
IfcElectricDistributionBoard			
IfcElectricDistributionBoardType			
IfcElectricFlowStorageDeviceType			
IfcElectricGenerator			
IfcElectricGeneratorType			
IfcElectricMotor			
IfcElectricMotorType			
IfcJunctionBox			
IfcJunctionBoxType			
IfcLamp			
IfcLampType			
IfcLightFixture			
IfcLightFixtureType			
IfcOutlet			
IfcOutletType			
IfcProtectiveDevice			
IfcProtectiveDeviceType			
IfcSolarDevice			

Entity	Source	Name	Tokens
IfcSolarDeviceType			
IfcSwitchingDevice			
IfcSwitchingDeviceType			
IfcTransformer			
IfcTransformerType			
IfcChiller			
IfcChillerType			
IfcFlowMeter			
IfcFlowMeterType			
IfcPump			
IfcPumpType			
IfcUnitaryEquipment			
IfcUnitaryEquipmentType			

Placement

Entity	Type
IfcBuilding	
IfcGrid	
IfcSite	
IfcSpace	
IfcTransportElement	
IfcCovering	
IfcSlabElementedCase	
IfcWallElementedCase	
IfcDistributionCircuit	
IfcDistributionPort	
IfcCableCarrierFitting	
IfcCableCarrierSegment	
IfcCableFitting	
IfcCableSegment	IfcLocalPlacement
IfcElectricAppliance	
IfcElectricDistributionBoard	
IfcElectricFlowStorageDevice	
IfcElectricGenerator	
IfcElectricMotor	

Entity	Type
IfcJunctionBox	
IfcLamp	
IfcLightFixture	IfcGridPlacement
	IfcLocalPlacement
IfcOutlet	
IfcProtectiveDevice	
IfcSolarDevice	
IfcSolarDeviceType	
IfcSwitchingDevice	
IfcTransformer	
IfcChiller	
IfcFlowMeter	
IfcPump	IfcLocalPlacement
IfcUnitaryEquipment	

Element Decomposition

Entity	RelatedObjects
IfcBuildingStorey	IfcSpace
IfcCovering	IfcGrid
	IfcLightFixture
IfcElectricDistributionBoard	IfcProtectiveDevice

Object Typing

Entity	Type
IfcDistributionElement	IfcDistributionElementType
IfcSpace	IfcSpaceType
IfcTransportElement	
IfcConstructionEquipmentResource	IfcConstructionEquipmentResourceType
IfcSubContractResource	
IfcCableCarrierFitting	IfcCableCarrierFittingType
IfcCableCarrierSegment	IfcCableCarrierSegmentType
IfcCableFitting	IfcCableFittingType
IfcCableSegment	IfcCableSegmentType
IfcElectricAppliance	IfcElectricApplianceType
IfcElectricDistributionBoard	IfcElectricDistributionBoardType
IfcElectricFlowStorageDevice	IfcElectricFlowStorageDeviceType
IfcElectricGenerator	IfcElectricGeneratorType

IfcElectricMotor	IfcElectricMotorType
IfcJunctionBox	IfcJunctionBoxType
IfcLamp	IfcLampType
IfcLightFixture	IfcLightFixtureType
IfcOutlet	IfcOutletType
IfcProtectiveDevice	IfcProtectiveDeviceType
IfcSolarDevice	IfcSolarDeviceType
IfcSwitchingDevice	IfcSwitchingDeviceType
IfcTransformer	IfcTransformerType
IfcChiller	IfcChillerType
IfcFlowMeter	IfcFlowMeterType
IfcPump	IfcPumpType
IfcUnitaryEquipment	IfcUnitaryEquipmentType

Footprint Geometry

Entity	RepresentationType	Geometry
IfcDistributionElement	GeometricCurveSet	IfcGeometricCurveSet
IfcSite	GeometricCurveSet	IfcGeometricCurveSet
	Annotation2D	IfcAnnotationFillArea
IfcCovering		
IfcSlabElementedCase		
IfcDistributionCircuit		
IfcElectricGenerator		
IfcElectricMotor		
IfcElectricMotorType		
IfcLightFixture		
IfcTransformer		
IfcChiller		
IfcFlowMeter		
IfcPump		
IfcUnitaryEquipment		

Body Geometry

Entity	RepresentationType	Geometry
IfcDistributionElement	Brep	IfcFacetedBrep
	SurfaceModel	IfcFaceBasedSurfaceModel
	SweptSolid	IfcExtrudedAreaSolid
	SectionedSpine	IfcSectionedSpine
	CSG	IfcCsgSolid

Entity	RepresentationType	Geometry
	Tessellation	IfcTriangulatedFaceSet
	MappedRepresentation	IfcMappedItem
IfcSite		
IfcSpace		
IfcTransportElement	SurfaceModel	IfcFaceBasedSurfaceModel
	Brep	IfcFacetedBrep
	Tessellation	IfcTriangulatedFaceSet
	MappedRepresentation	IfcMappedItem
IfcCableCarrierFitting		
IfcCableCarrierSegment		
IfcCableFitting		
IfcCableSegment		
IfcElectricAppliance		
IfcElectricApplianceType		
IfcElectricDistributionBoard		
IfcElectricDistributionBoardType		
IfcElectricFlowStorageDevice		
IfcElectricGenerator		
IfcElectricMotor		
IfcElectricMotorType		
IfcJunctionBox		
IfcLamp		
IfcLightFixture		
IfcOutlet		
IfcProtectiveDevice		
IfcSolarDevice		
IfcSwitchingDevice		
IfcTransformer		
IfcChiller		
IfcFlowMeter		
IfcPump		
IfcUnitaryEquipment		

Spatial Containment

Entity	Structure
--------	-----------

Entity	Structure
IfcGrid	
IfcTransportElement	
IfcCovering	IfcSite
	IfcBuildingStorey
	IfcBuildingStorey
IfcSlabElementedCase	IfcSite
	IfcBuildingStorey
	IfcBuildingStorey
IfcWallElementedCase	IfcSite
	IfcBuildingStorey
	IfcBuildingStorey
IfcCableCarrierFitting	IfcSite
	IfcBuildingStorey
	IfcBuildingStorey
IfcCableCarrierSegment	IfcSite
	IfcBuildingStorey
IfcCableFitting	
IfcCableSegment	IfcSite
	IfcBuildingStorey
IfcElectricAppliance	
IfcElectricDistributionBoard	IfcSpace
IfcElectricFlowStorageDevice	IfcSite
	IfcBuildingStorey
	IfcBuildingStorey
IfcElectricGenerator	IfcSpace
IfcElectricMotor	
IfcJunctionBox	
IfcLamp	
IfcLightFixture	
IfcOutlet	
IfcProtectiveDevice	
IfcSolarDevice	IfcSpace
IfcSwitchingDevice	
IfcTransformer	IfcSpace
IfcChiller	
IfcFlowMeter	IfcSite
IfcPump	

Entity	Structure
IfcUnitaryEquipment	

Grid

Entity	GridType	UCurve	VCurve
IfcGrid	RECTANGULAR	IfcLine	IfcLine

Space Coverings

Entity	CoveringType
IfcSpace	CEILING

Properties for Types

Entity	Name
IfcSpaceType	Pset_SpaceCommon
	Pset_SpaceOccupancyRequirements
	Pset_SpaceLightingRequirements
IfcTransportElementType	
IfcCableCarrierFitting	
IfcCableCarrierFittingType	
IfcCableCarrierSegmentType	
IfcCableFitting	
IfcCableSegmentType	
IfcElectricApplianceType	
IfcElectricDistributionBoardType	
IfcElectricFlowStorageDeviceType	
IfcElectricGeneratorType	
IfcElectricMotorType	
IfcJunctionBoxType	
IfcLampType	
IfcLightFixtureType	
IfcOutletType	
IfcProtectiveDeviceType	
IfcSolarDeviceType	
IfcSwitchingDeviceType	
IfcTransformerType	
IfcChiller	
IfcChillerType	

Entity	Name
IfcFlowMeterType	
IfcPumpType	
IfcUnitaryEquipment	
IfcUnitaryEquipmentType	

Ports

Entity	PredefinedType	Name	Flow	Type
IfcTransportElement				
IfcCableCarrierFitting	BEND	Head	SINK	NOTDEFINED
	BEND	Tail	SOURCE	NOTDEFINED
	CROSS	Head	SINK	NOTDEFINED
	CROSS	Tail	SOURCE	NOTDEFINED
	CROSS	Left	SOURCE	NOTDEFINED
	CROSS	Right	SOURCE	NOTDEFINED
	REDUCER	Head	SINK	NOTDEFINED
	REDUCER	Tail	SOURCE	NOTDEFINED
	TEE	Head	SINK	NOTDEFINED
	TEE	Left	SOURCE	NOTDEFINED
	TEE	Right	SOURCE	NOTDEFINED
IfcCableCarrierFittingType	BEND	Head	SINK	NOTDEFINED
	BEND	Tail	SOURCE	NOTDEFINED
	CROSS	Head	SINK	NOTDEFINED
	CROSS	Tail	SOURCE	NOTDEFINED
	CROSS	Left	SOURCE	NOTDEFINED
	CROSS	Right	SOURCE	NOTDEFINED
	REDUCER	Head	SINK	NOTDEFINED
	REDUCER	Tail	SOURCE	NOTDEFINED
	TEE	Head	SINK	NOTDEFINED
	TEE	Left	SOURCE	NOTDEFINED
	TEE	Right	SOURCE	NOTDEFINED
IfcCableCarrierSegment		Head	SINK	NOTDEFINED
		Tail	SOURCE	NOTDEFINED
IfcCableCarrierSegmentType		Head	SINK	NOTDEFINED
		Tail	SOURCE	NOTDEFINED
IfcCableFitting	CONNECTOR	Input	SINK	NOTDEFINED
	CONNECTOR	Output	SOURCE	NOTDEFINED
	ENTRY	Output	SOURCE	NOTDEFINED
	EXIT	Input	SINK	NOTDEFINED

Entity	PredefinedType	Name	Flow	Type
	JUNCTION	Input	SINK	NOTDEFINED
	JUNCTION	Output#1	SOURCE	NOTDEFINED
	JUNCTION	Output#2	SOURCE	NOTDEFINED
	CONNECTOR	Input	SINK	NOTDEFINED
	CONNECTOR	Output	SOURCE	NOTDEFINED
IfcCableSegment		Head	SINK	ELECTRICAL
		Tail	SOURCE	ELECTRICAL
IfcCableSegmentType		Head	SINK	ELECTRICAL
		Tail	SOURCE	ELECTRICAL
IfcElectricAppliance	DISHWASHER	Power	SINK	ELECTRICAL
	DISHWASHER	HotWater	SINK	DOMESTICHOTWATER
	DISHWASHER	Drainage	SOURCE	DRAINAGE
	ELECTRICCOOKER	Power	SINK	ELECTRICAL
	FREEZER	Power	SINK	ELECTRICAL
	FRIDGE_FREEZER	Power	SINK	ELECTRICAL
	FRIDGE_FREEZER	ColdWater	SINK	DOMESTICCOLDWATER
	HANDDRYER	Power	SINK	ELECTRICAL
	MICROWAVE	Power	SINK	ELECTRICAL
	REFRIGERATOR	Power	SINK	ELECTRICAL
	TUMBLEDRYER	Power	SINK	ELECTRICAL
	TUMBLEDRYER	Gas	SINK	GAS
	TUMBLEDRYER	Exhaust	SINK	EXHAUST
	WASHINGMACHINE	Power	SINK	ELECTRICAL
	WASHINGMACHINE	ColdWater	SINK	DOMESTICCOLDWATER
	WASHINGMACHINE	HotWater	SINK	DOMESTICHOTWATER
	WASHINGMACHINE	Drainage	SOURCE	DRAINAGE
IfcElectricDistributionBoard	CONSUMERUNIT	Line	SINK	ELECTRICAL
	CONSUMERUNIT	Ground	SINK	EARTHING
	CONSUMERUNIT	Circuit#1	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#2	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#3	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#4	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#5	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#6	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#7	SOURCE	ELECTRICAL
	CONSUMERUNIT	Circuit#8	SOURCE	ELECTRICAL
IfcElectricFlowStorageDevice		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcElectricFlowStorageDeviceType		Line	SINK	ELECTRICAL

Entity	PredefinedType	Name	Flow	Type
		Load	SOURCE	ELECTRICAL
IfcElectricGenerator		Load	SOURCE	ELECTRICAL
IfcElectricGeneratorType		Load	SOURCE	ELECTRICAL
IfcElectricMotor		Line	SINK	ELECTRICAL
		Drive	SOURCE	NOTDEFINED
IfcElectricMotorType		Line	SINK	ELECTRICAL
		Drive	SOURCE	NOTDEFINED
IfcJunctionBox	DATA	Line#1	SINK	DATA
	DATA	Line#2	SINK	DATA
	DATA	Gang#1	SOURCE	DATA
	DATA	Gang#2	SOURCE	DATA
	POWER	Line	SINK	ELECTRICAL
	POWER	Load	SOURCE	ELECTRICAL
	POWER	Gang#1	SOURCE	ELECTRICAL
	POWER	Gang#2	SOURCE	ELECTRICAL
	POWER	Gang#3	SOURCE	ELECTRICAL
	POWER	Gang#4	SOURCE	ELECTRICAL
IfcJunctionBoxType	DATA	Line#1	SINK	DATA
	DATA	Line#2	SINK	DATA
	DATA	Gang#1	SOURCE	DATA
	DATA	Gang#2	SOURCE	DATA
	POWER	Line	SINK	ELECTRICAL
	POWER	Load	SOURCE	ELECTRICAL
	POWER	Gang#1	SOURCE	ELECTRICAL
	POWER	Gang#2	SOURCE	ELECTRICAL
	POWER	Gang#3	SOURCE	ELECTRICAL
	POWER	Gang#4	SOURCE	ELECTRICAL
IfcLamp				
IfcLightFixture	POINTSOURCE	Line	SINK	ELECTRICAL
	POINTSOURCE	Socket	SOURCE	LIGHTING
	DIRECTIONSOURCE	Line	SINK	ELECTRICAL
	DIRECTIONSOURCE	Socket#1	SOURCE	LIGHTING
	DIRECTIONSOURCE	Socket#2	SOURCE	LIGHTING
	DIRECTIONSOURCE	Socket#3	SOURCE	LIGHTING
	DIRECTIONSOURCE	Socket#4	SOURCE	LIGHTING
IfcOutlet	DATAOUTLET	Line#1	SINK	DATA
	DATAOUTLET	Line#2	SINK	DATA
	DATAOUTLET	Jack#1	SOURCE	DATA

Entity	PredefinedType	Name	Flow	Type
	DATAOUTLET	Jack#2	SOURCE	DATA
	POWEROUTLET	Line#1	SINK	ELECTRICAL
	POWEROUTLET	Jack#1	SOURCE	ELECTRICAL
	POWEROUTLET	Jack#1	SOURCE	ELECTRICAL
	TELEPHONEOUTLET	Line#1	SINK	TELEPHONE
	TELEPHONEOUTLET	Line#2	SINK	TELEPHONE
	TELEPHONEOUTLET	Jack#1	SINK	TELEPHONE
	TELEPHONEOUTLET	Jack#2	SINK	TELEPHONE
IfcOutletType	DATAOUTLET	Line#1	SINK	DATA
	DATAOUTLET	Line#2	SINK	DATA
	DATAOUTLET	Jack#1	SOURCE	DATA
	DATAOUTLET	Jack#2	SOURCE	DATA
	POWEROUTLET	Line#1	SINK	ELECTRICAL
	POWEROUTLET	Jack#1	SOURCE	ELECTRICAL
	POWEROUTLET	Jack#1	SOURCE	ELECTRICAL
	TELEPHONEOUTLET	Line#1	SINK	TELEPHONE
	TELEPHONEOUTLET	Line#2	SINK	TELEPHONE
	TELEPHONEOUTLET	Jack#1	SINK	TELEPHONE
	TELEPHONEOUTLET	Jack#2	SINK	TELEPHONE
IfcProtectiveDevice		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcProtectiveDeviceType		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcSolarDevice	SOLARCOLLECTOR	Inlet	SOURCE	HEATING
	SOLARCOLLECTOR	Outlet	SINK	HEATING
	SOLARPANEL	Load	SOURCE	POWERGENERATION
IfcSwitchingDevice		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcSwitchingDeviceType		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcTransformer		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcTransformerType		Line	SINK	ELECTRICAL
		Load	SOURCE	ELECTRICAL
IfcChiller	AIRCOOLED	Power	SINK	ELECTRICAL
	AIRCOOLED	Control	SINK	CONTROL
	AIRCOOLED	ChilledWaterIn	SINK	CHILLEDWATER
	AIRCOOLED	ChilledWaterOut	SOURCE	CHILLEDWATER
	AIRCOOLED	VentilationIn	SINK	VENTILATION

Entity	PredefinedType	Name	Flow	Type
	AIRCOOLED	VentilationOut	SOURCE	VENTILATION
	WATERCOOLED	Power	SINK	ELECTRICAL
	WATERCOOLED	Control	SINK	CONTROL
	WATERCOOLED	ChilledWaterIn	SINK	CHILLEDWATER
	WATERCOOLED	ChilledWaterOut	SOURCE	CHILLEDWATER
	WATERCOOLED	CondenserWaterIn	SINK	CONDENSERWATER
	WATERCOOLED	CondenserWaterOut	SOURCE	CONDENSERWATER
IfcFlowMeter	ENERGYMETER	Inlet	SINK	ELECTRICAL
	ENERGYMETER	Outlet	SOURCE	ELECTRICAL
	GASMETER	Inlet	SINK	GAS
	GASMETER	Outlet	SOURCE	GAS
	OILMETER	Inlet	SINK	OIL
	OILMETER	Outlet	SOURCE	OIL
	WATERMETER	Inlet	SINK	DOMESTICCOLDWATER
	WATERMETER	Outlet	SOURCE	DOMESTICCOLDWATER
IfcPump		Power	SINK	ELECTRICAL
		Inlet	SINK	NOTDEFINED
		Outlet	SOURCE	NOTDEFINED
IfcUnitaryEquipment	AIRHANDLER	ReturnAirIn	SINK	AIRCONDITIONING
	AIRHANDLER	SupplyAirOut	SOURCE	AIRCONDITIONING
	AIRHANDLER	OutsideAirIn	SINK	VENTILATION
	AIRHANDLER	ExhaustAirOut	SOURCE	EXHAUST
	AIRHANDLER	ChilledWaterIn	SINK	CHILLEDWATER
	AIRHANDLER	ChilledWaterOut	SOURCE	CHILLEDWATER
	AIRHANDLER	HeatingIn	SINK	HEATING
	AIRHANDLER	HeatingOut	SOURCE	HEATING
	AIRHANDLER	Power	SINK	ELECTRICAL
	AIRHANDLER	Control	SINK	CONTROL

Product Type Assignment

Entity
IfcTransportElementType
IfcCableCarrierFittingType
IfcCableCarrierSegmentType
IfcCableSegmentType
IfcElectricApplianceType
IfcElectricDistributionBoardType
IfcElectricFlowStorageDeviceType
IfcElectricGeneratorType

IfcElectricMotorType
IfcJunctionBoxType
IfcLampType
IfcLightFixtureType
IfcOutletType
IfcProtectiveDeviceType
IfcSolarDeviceType
IfcSwitchingDeviceType
IfcTransformerType
IfcChillerType
IfcFlowMeterType
IfcPumpType
IfcUnitaryEquipmentType

Surface Geometry

Entity	RepresentationType	Geometry
IfcSlabElementedCase	Surface3D	IfcBoundedSurface
IfcWallElementedCase	Surface3D	IfcBoundedSurface

Axis Geometry

Entity	RepresentationType	Geometry
IfcWallElementedCase	Curve2D	IfcBoundedCurve
IfcCableCarrierSegment	Curve2D	IfcBoundedCurve
IfcCableSegment	Curve2D	IfcBoundedCurve

Group Assignment

Entity	Type
IfcDistributionCircuit	IfcJunctionBox
	IfcChiller
	IfcUnitaryEquipment
	IfcPump
	IfcLightFixture
IfcDistributionSystem	IfcElectricDistributionBoard

Material Profile Set Usage

Entity	Name
IfcDistributionPort	
IfcCableCarrierSegment	Casing
IfcCableSegment	Conductor

	Insulation
	Screen
	Sheath

Resource Cost

Entity	CostType	CostName	ValueType
IfcConstructionEquipmentResource			
IfcSubContractResource			

Resource Quantity

Entity	QuantityType	QuantityName
IfcConstructionEquipmentResource		
IfcSubContractResource		

Resource Assignment

Entity	Type
IfcSubContractResource	IfcActor

Clearance Geometry

Entity	RepresentationType	Geometry
IfcElectricApplianceType		
IfcElectricDistributionBoard		
IfcElectricDistributionBoardType		
IfcElectricGenerator		
IfcElectricMotor		
IfcElectricMotorType		
IfcTransformer		
IfcChiller		
IfcFlowMeter		
IfcPump		
IfcUnitaryEquipment		

Lighting Geometry

Entity	Type	Geometry
IfcLamp		
IfcLightFixture	LightSource	IfcLightSource

4.7.6.2.4 Concept relationship description

Concepts may inherit from other concepts such that more generic rules may be defined at a higher level

and more specific rules at a lower level. For example, geometry may be defined for a distribution segment (e.g. ducts, pipes, cables) that indicate permitted use of an extruded area solid (IfcExtrudedAreaSolid) which defines a 2D cross section extruded along a 3D linear segment. Such rule may be further refined for ducts to indicate that the cross-sections are further restricted to shapes such as hollow rectangles (IfcRectangleHollowProfileDef) or hollow circles (IfcCircleHollowProfileDef). Concepts are shown in a hierarchy as follows where inner concepts inherit from outer concepts.

- Roots
 - Identity
 - Revision Control
 - Descriptions
- Project
 - Project Declaration
 - Project Units
 - Conversion Units
 - Project Context
- Association
 - Classification
 - Material
 - Material Profile Set
 - Material Profile Set Usage
- Definition
 - Object Typing
 - Property Sets
 - Properties for Occurrences
 - Properties for Types
 - Properties for Performance
- Assignment
 - Actor Assignment
 - Control Assignment
 - Group Assignment
 - Process Assignment
 - Resource Assignment
 - Product Type Assignment
 - Process Type Assignment
- Composition
 - Aggregation
 - Element Decomposition
 - Spatial Decomposition
 - Nesting
 - Ports
 - Type-Based Ports
- Connectivity
 - Spatial Structure
 - Spatial Containment
 - Space Coverings
 - Port Connectivity
 - Sequential Connectivity
 - Interference

- Actor
 - Organization Role
- Control
 - Cost
 - Calendar
- Product
 - Placement
 - Geometry
 - Axis Geometry
 - Axis 2D Geometry
 - Axis 3D Geometry
 - Footprint Geometry
 - FootPrint GeomSet Geometry
 - FootPrint Annotation Geometry
 - Surface Geometry
 - Surface 3D Geometry
 - Body Geometry
 - Body SurfaceOrSolidModel Geometry
 - Body SurfaceModel Geometry
 - Body Tessellation Geometry
 - Body SweptSolid Geometry
 - Body AdvancedSweptSolid Geometry
 - Body Brep Geometry
 - Body AdvancedBrep Geometry
 - Body CSG Geometry
 - Body Clipping Geometry
 - Clearance Geometry
 - Lighting Geometry
 - Site Location
 - Building Location
 - Grid
- Process
 - Task Scheduling
- Resource
 - Resource Cost
 - Resource Quantity
- Resource Type
 - Resource Cost Rate

4.7.6.2.5 Concept requirements applicability

Each entity is shown in subsections as follows, with rows corresponding to concepts, columns corresponding to exchanges, and cells indicating requirements where 'R' indicates required and 'O' indicates optional.

IfcActor

Design Coordinated				
System Operation				
Product Type Selection				
Product Template				
Product Type Template				
Design Schematic				
System Layout				
Design Early				
Product Program				
Space Program				
Project Definition				
Discipline Specifications				
Facility Criteria				
Concept				
Identity				
Properties for Occurrences				
Organization Role				
Actor Assignment				

IfcProject

Design Coordinated												
System Operation	R	O	O	O	R	R	R	R	R	R	R	R
Product Type Selection	R	O	O	O	R	R	R	R	R	R	R	R
Product Template	R	O	O	O	R	R	R	R	R	R	R	R
Product Type Template	R	O	O	O	R	R	R	R	R	R	R	R
Design Schematic	R	O	O	O	R	R	R	R	R	R	R	R
System Layout	R	O	O	O	R	R	R	R	R	R	R	R
Design Early	R	O	O	O	R	R	R	R	R	R	R	R
Product Program	R	O	O	O	R	R	R	R	R	R	R	R
Space Program	R	O	O	O	R	R	R	R	R	R	R	R
Project Definition	R	O	O	O	R	R	R	R	R	R	R	R
Discipline Specifications	R	O	O	O	R	R	R	R	R	R	R	R
Facility Criteria	R	O	O	O	R	R	R	R	R	R	R	R
Concept	R	O	O	O	R	R	R	R	R	R	R	R
Identity	R	O	O	O	R	R	R	R	R	R	R	R
Revision Control	O	O	O	O	O	O	O	O	O	O	O	O
Descriptions	O	O	O	O	O	O	O	O	O	O	O	O
Spatial Decomposition	R	R	R	R	R	R	R	R	R	R	R	R
Conversion Units	O	O	O	O	O	O	O	O	O	O	O	O
Project Context	R	R	R	R	R	R	R	R	R	R	R	R
Project Declaration	O	O	O	O	O	O	O	O	O	O	O	O

IfcPerformanceHistory

Design Coordinated												
System Operation												
Product Type Selection												
Product Template												
Product Type Template												
Design Schematic												
System Layout												
Design Early												
Product Program												
Space Program												
Project Definition												
Discipline Specifications												
Facility Criteria												
Concept												

IfcGrid

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R										
Spatial Containment	R	R	R										
Placement	R	R	R										
Grid	R	R	R										

IfcSite

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R										
Site Location	R	R	R										
Spatial Decomposition	R	R	R										
Footprint Geometry	R	R	R										
Body Geometry	R	R	R										
Placement	R	R	R										

IfcSpace

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R	R	R	R	R		
Object Typing	R	R	R			R	R	R	R	R	R	R		
Properties for Occurrences	R	R	R			R	R	R	R	R	R	O		
Placement	R	R	R			R	R	R	R	R	R	O		
Space Coverings	R	R	R			R	R	R	R	R	R	O		
Classification	R	R	R			R	R	R	R	R	R	O		
Body Geometry	R	R	R			R	R	R	R	R	R	O		

IfcSpaceType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity														
Classification													R	
Properties for Types													R	

IfcTransportElement

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R				R	R	R	R		R		
Object Typing	R	R	R			R	R	R	R	R		R		
Properties for Occurrences	R	R	R			R	R	R	R	R		R		
Placement	R	R	R			R	R	R	R	R		R		
Body Geometry	R	R	R			R	R	R	R	R		R		
Ports	R	R	R			R	R	R	R	R		R		
Spatial Containment		R				R	R	R	O	R		R		
Classification	R					R	R	R	O	R		R		

IfcTransportElementType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									
Product Type Assignment	R	R	O	R	R									
Type-Based Ports	R	R	R	R	R									
Classification	R	R	R	R	R									

IfcCovering

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity						R								
Element Decomposition						R								
Spatial Containment						R								
Placement						R								
Footprint Geometry						R								

IfcSlabElementedCase

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline	Facility Criteria	Concept
Identity	R	R	R			R	R	R						
Placement	R	R	R			R	R	R						
Surface Geometry	R	R	R			R	R	R						
Spatial Containment	R	R	R			R	R	R						
Aggregation	R	R	R			R	R	R						
Footprint Geometry	R	R	R			R	R	R						
Axis Geometry	R	R	R			R	R	R						

IfcWallElementedCase

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R						
Placement	R	R	R			R	R	R						
Axis Geometry	R	R	R			R	R	R						
Surface Geometry	R	R	R			R	R	R						
Spatial Containment	R	R	R			R	R	R						
Aggregation	R	R	R			R	R	R						

IfcDistributionCircuit

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R		R			R	R							
Properties for Occurrences	R		R			R	R							
Group Assignment	R	R	R			R	R							

IfcDistributionPort

Design Coordinated												
System Operation												
Product Type Selection												
Product Template												
Product Type Template												
Design Schematic												
System Layout												
Design Early												
Product Program												
Space Program												
Project Definition												
Discipline Specifications												
Facility Criteria												
Concept												
Identity												
Properties for Occurrences												
Material Profile Set Usage												
Port Connectivity												
Placement												

IfcDistributionSystem

Design Coordinated												
System Operation												
Product Type Selection												
Product Template												
Product Type Template												
Design Schematic												
System Layout												
Design Early												
Product Program												
Space Program												
Project Definition												
Discipline Specifications												
Facility Criteria												
Concept												
Identity												
Properties for Occurrences												
Aggregation												
Group Assignment												

IfcOccupant

Design Coordinated												
System Operation												
Product Type Selection												
Product Template												
Product Type Template												
Design Schematic												
System Layout												
Design Early												
Product Program												
Space Program												
Project Definition												
Discipline Specifications												
Facility Criteria												
Concept												
Identity												
Actor Assignment												

Properties for Occurrences	R	R												
Organization Role	R	R												

IfcSubContractResource

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity											R		
Object Typing											R		
Resource Cost											R		
Resource Quantity											R		
Resource Assignment											R		

IfcSubContractResourceType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity											R		
Resource Cost Rate											R		

IfcCableCarrierFitting

Design Coordinated	System Operation	Product Type	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline	Facility Criteria	Concept

Design Coordinated	System Operation	Product Type	Product Template	Product Type	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline	Facility Criteria	Concept
Identity	R	R	R		R	R							
Object Typing	R	R	R		R	R							
Properties for Occurrences	R	R	R		R	R							
Ports	R	R	R		R	R							
Spatial Containment	R	R	R		R	R							
Body Geometry	R	R	R		R								
Placement	R	R	R		R								
Properties for Types	R	R	R		R								
Classification	R	R	R		R								

IfcCableCarrierFittingType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R									
Properties for Types	R	R	R	R									
Product Type Assignment	R	O	R	R									
Ports	R	O	R	R									
Type-Based Ports	R	R	R	R									
Classification	R	R	R	R									

IfcCableCarrierSegment

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R		R	R							
Object Typing	R	R	R		R	R							
Properties for Occurrences	R	R	R		R	R							
Material Profile Set Usage	R	R	R		O	R							
Placement	R	R	R		R	R							
Axis Geometry	R	R	R		R	R							
Body Geometry	R	R	R										
Spatial Containment	R	R	R		O	R							
Interference	R	R	R		O	R							
Ports	R	R	R		R	R							
Classification	R	R	R		R	R							

IfcCableCarrierSegmentType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R								
Properties for Types	R	R	R	R	R								
Product Type Assignment	R	O	R	R	R								
Material Profile Set	R	R	R	R	R								
Ports	R	R	R	R	R								
Type-Based Ports	R	R	R	R	R								
Classification	R	R	R	R	R								

IfcCableFitting

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity					R		R	R			R	R	R
Object Typing					R		R	R			R	R	R
Properties for Occurrences					R		R	R			R	R	R
Ports							R	R			R	R	R
Spatial Containment							R	R			R	R	R
Body Geometry							R	R			R	R	R
Placement							R	R			R	R	R
Properties for Types							R	R			R	R	
Classification							R	R			R	R	R

IfcCableSegment

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity							R	R			R	R	R
Object Typing							R	R			R	R	R
Properties for Occurrences							R	R			R	R	R

Material Profile Set Usage							R	R			R	R	R
Spatial Containment							R	R			R	R	R
Placement							R	R			R	R	R
Ports							R	R			R	R	R
Axis Geometry								R			R	R	R
Body Geometry								R			R	R	R
Classification								R			R	R	R

IfcCableSegmentType

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity									R	R	R	R	R
Properties for Types									R	R	R	R	R
Product Type Assignment									R	R	O	R	
Material Profile Set									R	R	R	R	R
Ports									R	R	R	R	R
Type-Based Ports									R	R	R	R	R
Classification									R	R	R	R	R

IfcElectricAppliance

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity					R		R				R	R	R
Object Typing					R		R				R	R	R
Properties for Occurrences					R		R				R	R	R

Ports					R		R				R	R	R
Spatial Containment					O		R				R	R	R
Body Geometry					O		R				R	R	R
Placement					O		R				R	R	R
Classification					O		R				R	R	R

IfcElectricApplianceType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									
Type-Based Ports	R	R	R	R	R									
Body Geometry	R	R	R	R	R									
Clearance Geometry	R	R	R	R	R									
Classification	R	R	R	R	R									
Product Type Assignment	R	R	O	R	R									

IfcElectricDistributionBoard

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept

Identity					R	R	R	R			R	R	R
Object Typing					R	R	R	R			R	R	R
Properties for Occurrences					R	R	R	R			R	R	R
Body Geometry					R	R	R	R			R	R	R
Clearance Geometry					R	R	R	R			R	R	R
Element Decomposition					R	R	R	R			R	R	R
Ports					R	R	R	R			R	R	R
Spatial Containment					O	R	R	R			R	R	R
Placement					O	R	R	R			R	R	R
Classification					O	R	R	R			R	R	R

IfcElectricDistributionBoardType

	Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity										R	R	R	R	R
Properties for Types										R	R	R	R	R
Product Type Assignment										R	R	O	R	R
Body Geometry										R	R	O	R	R
Clearance Geometry										R	R	O	R	R
Type-Based Ports										R	R	R	R	R
Classification										R	R	R	R	R

IfcElectricFlowStorageDevice

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R					
Object Typing	R	R	R			R	R	R					
Properties for Occurrences	R	R	R			R	R	R					
Ports	R	R	R			R	R	R					
Spatial Containment	R	R	R			O	R	R					
Placement	R	R	R			O	R	R					
Body Geometry	R	R	R			O	R	R					

IfcElectricFlowStorageDeviceType

Design Coordinated	System Operation	Product Type	Product Template	Product Type	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline	Facility Criteria	Concept
Identity	R	R	R	R									
Properties for Types	R	R	R	R									
Product Type Assignment	R	O	R	R									
Ports	R	O	R	R									
Type-Based Ports	R	R	R	R									
Classification	R	R	R	R									

IfcElectricGenerator

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R					R	R	R			
Object Typing	R	R	R					R	R	R			
Properties for Occurrences	R	R	R					R	R	R			
Ports	R	R	R					R	R	R			
Footprint Geometry			O			O	O		O				
Body Geometry	R	R	R					O					
Clearance Geometry	R	R	R			R	R	O	R				
Spatial Containment	R	R	R			R	R	O	R				
Placement	R	R	R			R	R	O	R				
Classification	R	R	R			R	R	O	R				

IfcElectricGeneratorType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R									
Properties for Types	R	R	R	R									
Product Type Assignment	R	O	R	R									
Ports	R	O	R	R									

Type-Based Ports									R	R	R	R	R
Classification									R	R	R	R	R

IfcElectricMotor

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity		R	R	R			R	R	R	R				
Object Typing		R	R	R			R	R	R	R				
Properties for Occurrences		R	R	R			R	R	R	R				
Ports		R	R	R			R	R	R	R				
Footprint Geometry		R	R	R		O	R			R				
Body Geometry		R	R	R						R				
Clearance Geometry		R	R	R					O	R				
Spatial Containment		R	R	R		R			O	R				
Placement		R	R	R		R			O	R				
Classification		R	R	R		R			O	R				

IfcElectricMotorType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									

Ports										R	R	R	R	R
Footprint Geometry										R	R	R	R	R
Body Geometry										R	R	R	R	R
Clearance Geometry										R	R	R	R	R
Type-Based Ports										R	R	R	R	R
Classification										R	R	R	R	R
Product Type Assignment										R	R	O	R	R

IfcJunctionBox

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity							R	R			R	R	R
Object Typing							R	R			R	R	R
Properties for Occurrences							R	R			R	R	R
Ports							R	R			R	R	R
Spatial Containment							R	R			R	R	R
Placement							R	R			R	R	R
Body Geometry							R	R			R	R	R
Classification							R	R			R	R	R

IfcJunctionBoxType

IfcLamp

De
Sys
Proc
Proc
Proc
De
Sys
De
Proc
Spa
Proc
Dis
Fad
Col

IfcLampType

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity									R	R	R	R	R
Properties for Types									R	R	R	R	R
Product Type Assignment									R	R	O	R	R
Type-Based Ports									R	R	R	R	R
Classification									R	R	R	R	R

IfcLightFixture

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity					R	R	R	R			R	R	R
Object Typing					R	R	R	R			R	R	R
Properties for Occurrences					R	R	R	R			R	R	R
Placement					R	R		R			R	R	R
Ports					R	R	R	R			R	R	R
Spatial Containment					O	R	R	R			R	R	R
Body Geometry					O	R	R	R			R	R	R
Lighting Geometry					O	R	R				R	R	R
Classification					O	R	R				R	R	R
Footprint Geometry					O	R	R	O			R	R	R

IfcLightFixtureType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R									
Properties for Types	R	R	R	R									
Product Type Assignment	R	O	R	R									
Type-Based Ports	R	O	R	R									
Classification	R	O	R	R									

IfcOutlet

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R	R		R					
Object Typing	R	R	R	R	R	R		R					
Properties for Occurrences	R	R	R	R	R	R		R					
Ports	R	R	R	R	R	R		R					
Spatial Containment	R	R	R	R	R	R		O					
Body Geometry	R	R	R	R	R	R		O					
Placement	R	R	R	R	R	R		O					
Classification	R	R	R	R	R	R		O					

IfcOutletType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									
Product Type Assignment	R	R	O	R	R									
Ports	R	R	O	R	R									
Type-Based Ports	R	R	R	R	R									
Classification	R	R	R	R	R									

IfcProtectiveDevice

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R	R	R				
Object Typing	R	R	R			R	R	R	R	R				
Properties for Occurrences	R	R	R			R	R	R	R	R				
Ports	R	R	R			R	R	R	R	R				
Spatial Containment	R	R	R			R	R	R	O	O				
Body Geometry	R	R	R			R	R	R	O	O				
Placement	R	R	R			R	R	R	O	O				

Classification					O	R	R	R			R	R	R
----------------	--	--	--	--	---	---	---	---	--	--	---	---	---

IfcProtectiveDeviceType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									
Product Type Assignment	R	R	O	R	R									
Ports	R	R	O	R	R									
Type-Based Ports	R	R	R	R	R									
Classification	R	R	R	R	R									

IfcSolarDevice

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R								
Object Typing	R	R	R			R								
Properties for Occurrences	R	R	R			R								
Ports	R	R	R			R								
Spatial Containment	R	R	R			R								

Placement								R			R	R	R
Body Geometry								R			R	R	R
Classification								R			R	R	R

IfcSolarDeviceType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									
Type-Based Ports	R	R	R	R	R									
Classification	R	R	R	R	R									
Product Type Assignment	R	R	O	R	R									
Placement	R	R			R									

IfcSwitchingDevice

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R	R					
Object Typing	R	R	R			R	R	R	R					
Properties for Occurrences	R	R	R			R	R	R	R					

Ports					R	R	R	R			R	R	R
Spatial Containment					O	R	R	R			R	R	R
Body Geometry					O	R	R				R	R	R
Placement					O	R	R	R			R	R	R
Classification					O	R	R	R			R	R	R

IfcSwitchingDeviceType

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R	R									
Properties for Types	R	R	R	R	R									
Ports	R	R	R	R	R									
Type-Based Ports	R	R	R	R	R									
Classification	R	R	R	R	R									
Product Type Assignment	R	R	O	R	R									

IfcTransformer

	Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R	R					
Object Typing	R	R	R			R	R	R	R					

Properties for Occurrences					R	R	R	R			R	R	R
Ports					R	R	R	R			R	R	R
Spatial Containment					O	O	O	R			R	R	R
Clearance Geometry					O	O	O	R			R	R	R
Body Geometry					O	O	O	R			R	R	R
Placement					O	O	O	R			R	R	R
Classification					O	O	O	R			R	R	R
Footprint Geometry					O	O	O	O			R	R	R

IfcTransformerType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R									
Properties for Types	R	R	R	R									
Product Type Assignment	R	O	R	R									
Ports	R	O	R	R									
Type-Based Ports	R	R	R	R									
Classification	R	R	R	R									

IfcChiller

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept

Identity				R	R	R	R	R			R	R	R
Object Typing				R	R	R	R	R			R	R	R
Properties for Occurrences				R	R	R	R	R			R	R	R
Ports				R	R	R	R	R			R	R	R
Spatial Containment				R	O	R	R	R			R	R	R
Body Geometry				R	O	R	R	R			R	R	R
Placement				R	O	R	R	R			R	R	R
Classification				R	O	R	R	R			R	R	R
Properties for Types				R	O	R	R	R			R	R	
Footprint Geometry				R	O	R	R	O			R	R	
Clearance Geometry				R	O	R	R	R			R	R	

IfcChillerType

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity									R	R	R	R	R
Properties for Types									R	R	R	R	R
Type-Based Ports									R	R	R	R	R
Classification									R	R	R	R	R
Product Type Assignment									R	R	O	R	R

IfcFlowMeter

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R			R	R	R					
Object Typing	R	R	R			R	R	R					
Properties for Occurrences	R	R	R			R	R	R					
Ports	R	R	R			R	R	R					
Placement	R	R	R			R	R	R					
Spatial Containment	R	R	R			O	R	R					
Body Geometry	R	R	R			O	R	R					
Classification	R	R	R			O	R	R					
Footprint Geometry	R	R	R		O	R	R	R					
Clearance Geometry	R	R	R		R	R	R	R					

IfcFlowMeterType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R									
Properties for Types	R	R	R	R									
Type-Based Ports	R	R	R	R									
Classification	R	R	R	R									

Product Type Assignment									R	R	O	R	R
-------------------------	--	--	--	--	--	--	--	--	---	---	---	---	---

IfcPump

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R		R	R	R	R	R				
Object Typing	R	R	R		R	R	R	R	R				
Properties for Occurrences	R	R	R		R	R	R	R	R				
Ports	R	R	R		R	R	R	R	R				
Spatial Containment	R	R	R		R	R	R	O	R				
Body Geometry	R	R	R		R	R	R	O	R				
Placement	R	R	R		R	R	R	O	R				
Classification	R	R	R		R	R	R	O	R				
Footprint Geometry	R	R	R		O	R	R	O	R				
Clearance Geometry	R	R	R		R	R	R	O	R				

IfcPumpType

Design Coordinated	System Operation	Product Type Selection	Product Template	Product Type Template	Design Schematic	System Layout	Design Early	Product Program	Space Program	Project Definition	Discipline Specifications	Facility Criteria	Concept
Identity	R	R	R	R									

Properties for Types									R	R	R	R	R
Type-Based Ports									R	R	R	R	R
Classification									R	R	R	R	R
Product Type Assignment									R	R	O	R	R

IfcUnitaryEquipment

	Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity		R		R	R	R	R	R	R			R	R	R
Object Typing		R		R	R	R	R	R	R			R	R	R
Properties for Occurrences		O		R	R	R	R	R	R			R	R	R
Ports		O		R	R	R	R	R	R			R	R	R
Spatial Containment		O		R	O	R	R	R	R			R	R	R
Body Geometry		O		R	O	R	R	R	R			R	R	R
Placement		O		R	O	R	R	R	R			R	R	R
Classification		O		R	O	R	R	R	R			R	R	R
Properties for Types		O		R	O	R	R	R	R			R	R	
Footprint Geometry		O		R	O	R	R	O				R	R	
Clearance Geometry		O		R	O	R	R	R				R	R	

IfcUnitaryEquipmentType

Concept	Facility Criteria	Discipline Specifications	Project Definition	Space Program	Product Program	Design Early	System Layout	Design Schematic	Product Type Template	Product Template	Product Type Selection	System Operation	Design Coordinated
Identity									R	R	R	R	R
Properties for Types									R	R	R	R	R
Type-Based Ports									R	R	R	R	R
Classification									R	R	R	R	R
Product Type Assignment									R	R	O	R	R

4.7.6.3 Concept reusability

4.7.6.3.1 Concept list

Each concept used within this model view is listed as follows.

- Identity
- Properties for Occurrences
- Organization Role
- Actor Assignment
- Revision Control
- Descriptions
- Spatial Decomposition
- Conversion Units
- Project Context
- Project Declaration
- Properties for Performance
- Control Assignment
- Process Assignment
- Sequential Connectivity
- Task Scheduling
- Nesting
- Process Type Assignment
- Calendar

- Aggregation
- Classification
- Building Location
- Placement
- Element Decomposition
- Object Typing
- Footprint Geometry
- Body Geometry
- Spatial Containment
- Grid
- Site Location
- Space Coverings
- Properties for Types
- Ports
- Product Type Assignment
- Type-Based Ports
- Surface Geometry
- Axis Geometry
- Group Assignment
- Material Profile Set Usage
- Port Connectivity
- Cost
- Resource Cost
- Resource Quantity
- Resource Assignment
- Resource Cost Rate
- Interference
- Material Profile Set
- Clearance Geometry
- Lighting Geometry

4.7.6.3.2 Related existing concept list

In the following table, each row corresponds to a concept used within this model view, each column corresponds to another model view, and each cell indicates usage of the concept within the corresponding model view.

Concept	Common Use Definitions	Facilities Management Handover	Building Programming	Building Automation	Mechanical System Design	Electrical System Design	Plumbing System Design
Identity	X	X	X	X	X	X	X
Properties for Occurrences	X	X	X	X		X	X
Organization Role						X	X
Actor Assignment	X		X			X	X
Revision Control	X	X		X	X	X	X
Descriptions			X		X	X	X
Spatial Decomposition	X		X	X	X	X	X
Conversion Units	X	X		X	X	X	X
Project Context	X				X	X	X
Project Declaration	X	X	X	X	X	X	X
Properties for Performance	X			X		X	X
Control Assignment	X	X	X	X		X	X
Process Assignment	X					X	X
Sequential Connectivity	X	X				X	
Task Scheduling		X		X		X	
Nesting	X			X		X	X
Process Type Assignment						X	
Calendar			X			X	X
Aggregation	X			X	X	X	X
Classification	X	X	X	X	X	X	X
Building Location						X	X
Placement	X					X	X
Element Decomposition	X					X	
Object Typing	X	X	X	X	X	X	X
Footprint Geometry	X					X	X
Body Geometry	X					X	X

Concept	Common Use Definitions	Facilities Management Handover	Building Programming	Building Automation	Mechanical System Design	Electrical System Design	Plumbing System Design
Spatial Containment	X	X		X		X	X
Grid						X	
Site Location						X	X
Space Coverings						X	
Properties for Types		X	X	X		X	X
Ports	X			X		X	X
Product Type Assignment	X			X		X	
Type-Based Ports						X	X
Surface Geometry	X					X	X
Axis Geometry	X					X	X
Group Assignment	X	X		X		X	X
Material Profile Set Usage	X					X	X
Port Connectivity	X			X		X	X
Cost						X	
Resource Cost	X					X	X
Resource Quantity	X					X	X
Resource Assignment	X					X	X
Resource Cost Rate						X	X
Interference						X	X
Material Profile Set	X					X	X
Clearance Geometry	X					X	X
Lighting Geometry	X					X	

4.7.6.3.3 (not used)

4.7.6.3.4 Concept business rule list

Each concept template is defined in a subsection as follows, with rows corresponding to each business rule. The *Reference* column identifies the path to the entity and attribute. The *Cardinality* column indicates whether the number of permitted instances is restricted differently than the underlying schema, using [N:M] notation where N indicates the minimum number of instances, M indicates the maximum number of instances, where '?' indicates unbounded. The *Parameter* column indicates the name of a substitutable parameter, if applicable, defined at each usage of the business rule.

Identity

Reference	Cardinality	Parameter
\IfcRoot.GlobalId		
\IfcRoot.GlobalId\IfcGloballyUniqueId		
\IfcRoot.Name		
\IfcRoot.Name\IfcLabel		
\IfcRoot.Description		
\IfcRoot.Description\IfcText		

Properties for Occurrences

Reference	Cardinality	Parameter
\IfcObject.PredefinedType		PredefinedType
\IfcObject.IsDefinedBy		
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties		
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition		
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet		
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.Name		Name
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties		
\IfcObject.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcProperty		

Organization Role

Reference	Cardinality	Parameter
\IfcActor.TheActor		
\IfcActor.TheActor\IfcOrganization		
\IfcActor.TheActor\IfcOrganization.Identification		
\IfcActor.TheActor\IfcOrganization.Identification\IfcIdentifier		
\IfcActor.TheActor\IfcOrganization.Name		
\IfcActor.TheActor\IfcOrganization.Name\IfcLabel		
\IfcActor.TheActor\IfcOrganization.Roles		
\IfcActor.TheActor\IfcOrganization.Roles\IfcActorRole		
\IfcActor.TheActor\IfcOrganization.Roles\IfcActorRole.Role		Role

\IfcActor.TheActor\IfcOrganization.Roles\IfcActorRole.UserDefinedRole		UserDefinedRole
\IfcActor.TheActor\IfcOrganization.Addresses		
\IfcActor.TheActor\IfcOrganization.Addresses\IfcTelecomAddress		
\IfcActor.TheActor\IfcOrganization.Addresses\IfcTelecomAddress.ElectronicMailAddresses		
\IfcActor.TheActor\IfcOrganization.Addresses\IfcTelecomAddress.ElectronicMailAddresses\IfcLabel		

Actor Assignment

Reference	Cardinality	Parameter
\IfcActor.IsActingUpon		
\IfcActor.IsActingUpon\IfcRelAssignsToActor		
\IfcActor.IsActingUpon\IfcRelAssignsToActor.RelatedObjects		Type
\IfcActor.IsActingUpon\IfcRelAssignsToActor.RelatedObjects\IfcControl		

Revision Control

Reference	Cardinality	Parameter
\IfcRoot.OwnerHistory		
\IfcRoot.OwnerHistory\IfcOwnerHistory		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningUser		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningUser\IfcPersonAndOrganization		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningUser\IfcPersonAndOrganization.ThePerson		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningUser\IfcPersonAndOrganization.ThePerson\IfcPerson		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningUser\IfcPersonAndOrganization.TheOrganization		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningUser\IfcPersonAndOrganization.TheOrganization\IfcOrganization		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningApplication		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningApplication\IfcApplication		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningApplication\IfcApplication.ApplicationDeveloper		
\IfcRoot.OwnerHistory\IfcOwnerHistory.OwningApplication\IfcApplication.ApplicationDeveloper\IfcOrganization		
\IfcRoot.OwnerHistory\IfcOwnerHistory.State		
\IfcRoot.OwnerHistory\IfcOwnerHistory.State\IfcStateEnum		
\IfcRoot.OwnerHistory\IfcOwnerHistory.ChangeAction		
\IfcRoot.OwnerHistory\IfcOwnerHistory.ChangeAction\IfcChangeActionEnum		
\IfcRoot.OwnerHistory\IfcOwnerHistory.CreationDate		
\IfcRoot.OwnerHistory\IfcOwnerHistory.CreationDate\IfcTimeStamp		

Descriptions

Reference	Cardinality	Parameter
\IfcRoot.Description		
\IfcRoot.Description\IfcText		

Spatial Decomposition

Reference	Cardinality	Parameter
\IfcObjectDefinition.IsDecomposedBy		
\IfcObjectDefinition.IsDecomposedBy\IfcRelAggregates		
\IfcObjectDefinition.IsDecomposedBy\IfcRelAggregates.RelatedObjects		Spatial Parts
\IfcObjectDefinition.IsDecomposedBy\IfcRelAggregates.RelatedObjects\IfcSpatialElement		

Conversion Units

Reference	Cardinality	Parameter
\IfcContext.UnitsInContext		
\IfcContext.UnitsInContext\IfcUnitAssignment		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.UnitType		UnitType
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.UnitType\IfcUnitEnum		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.Name		Name
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.Name\IfcLabel		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.ValueComponent		ConversionType
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.ValueComponent\IfcReal		ConversionFactor
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.UnitComponent		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.UnitComponent\IfcSIUnit		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.UnitComponent\IfcSIUnit.Prefix		BaseUnitPrefix
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.UnitComponent\IfcSIUnit.Prefix\IfcSIPrefix		
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.UnitComponent\IfcSIUnit.Name		BaseUnitName
\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionFactor\IfcMeasureWithUnit.UnitComponent\IfcSIUnit.Name\IfcSIUnitName		

\IfcContext.UnitsInContext\IfcUnitAssignment.Units\IfcConversionBasedUnitWithOffset.ConversionOffset		ConversionOffset
--	--	------------------

Project Context

Reference	Cardinality	Parameter
\IfcContext.RepresentationContexts		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.ContextIdentifier		ContextIdentifier
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.ContextIdentifier\IfcLabel		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.ContextType		ContextType
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.ContextType\IfcLabel		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.CoordinateSpaceDimension		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.CoordinateSpaceDimension\IfcDimensionCount		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.WorldCoordinateSystem		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.WorldCoordinateSystem\IfcAxis2Placement3D		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.TrueNorth		
\IfcContext.RepresentationContexts\IfcGeometricRepresentationContext.TrueNorth\IfcDirection		

Project Declaration

Reference	Cardinality	Parameter
\IfcContext.Declares		
\IfcContext.Declares\IfcRelDeclares		
\IfcContext.Declares\IfcRelDeclares.RelatedDefinitions		Type
\IfcContext.Declares\IfcRelDeclares.RelatedDefinitions\IfcObjectDefinition		

Properties for Performance

Reference	Cardinality	Parameter
\IfcPerformanceHistory.IsDefinedBy		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference		

Reference	Cardinality	Parameter
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries.Values		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries.Values\IfcIrregularTimeSeriesValue		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries.Values\IfcIrregularTimeSeriesValue.TimeStamp		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries.Values\IfcIrregularTimeSeriesValue.TimeStamp\IfcDateTime		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries.Values\IfcIrregularTimeSeriesValue.ListValues		
\IfcPerformanceHistory.IsDefinedBy\IfcRelDefinesByProperties.RelatingPropertyDefinition\IfcPropertySet.HasProperties\IfcPropertyReferenceValue.PropertyReference\IfcIrregularTimeSeries.Values\IfcIrregularTimeSeriesValue.ListValues\IfcValue		

Control Assignment

Reference	Cardinality	Parameter
\IfcControl.Controls		
\IfcControl.Controls\IfcRelAssignsToControl		
\IfcControl.Controls\IfcRelAssignsToControl.RelatedObjects		Type
\IfcControl.Controls\IfcRelAssignsToControl.RelatedObjects\IfcObject		

Process Assignment

Reference	Cardinality	Parameter
\IfcProcess.OperatesOn		
\IfcProcess.OperatesOn\IfcRelAssignsToProcess		
\IfcProcess.OperatesOn\IfcRelAssignsToProcess.RelatedObjects		Type
\IfcProcess.OperatesOn\IfcRelAssignsToProcess.RelatedObjects\IfcResource		

Sequential Connectivity

Reference	Cardinality	Parameter
\IfcProcess.IsPredecessorTo		
\IfcProcess.IsPredecessorTo\IfcRelSequence		
\IfcProcess.IsPredecessorTo\IfcRelSequence.RelatedProcess		
\IfcProcess.IsPredecessorTo\IfcRelSequence.RelatedProcess\IfcProcess		
\IfcProcess.IsSuccessorFrom		
\IfcProcess.IsSuccessorFrom\IfcRelSequence		

\IfcProcess.IsSuccessorFrom\IfcRelSequence.RelatingProcess		
\IfcProcess.IsSuccessorFrom\IfcRelSequence.RelatingProcess\IfcProcess		

Task Scheduling

Reference	Cardinality	Parameter
\IfcTask.TaskTime		
\IfcTask.TaskTime\IfcTaskTimeRecurring		
\IfcTask.TaskTime\IfcTaskTimeRecurring.ScheduleStart		
\IfcTask.TaskTime\IfcTaskTimeRecurring.ScheduleStart\IfcDateTime		
\IfcTask.TaskTime\IfcTaskTimeRecurring.ScheduleFinish		
\IfcTask.TaskTime\IfcTaskTimeRecurring.ScheduleFinish\IfcDateTime		
\IfcTask.TaskTime\IfcTaskTimeRecurring.ScheduleDuration		
\IfcTask.TaskTime\IfcTaskTimeRecurring.ScheduleDuration\IfcDuration		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.RecurrenceType		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.RecurrenceType\IfcRecurrenceTypeEnum		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.DayComponent		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.DayComponent\IfcDayInMonthNumber		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.WeekdayComponent		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.WeekdayComponent\IfcDayInWeekNumber		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.MonthComponent		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.MonthComponent\IfcMonthInYearNumber		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.Position		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.Position\IfcInteger		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.Interval		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.Interval\IfcInteger		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.Occurrences		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.Occurrences\IfcInteger		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.TimePeriods		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.TimePeriods\IfcTimePeriod		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.StartTime		

Reference	Cardinality	Parameter
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.StartTime\IfcTime		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.EndTime		
\IfcTask.TaskTime\IfcTaskTimeRecurring.Recurrence\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.EndTime\IfcTime		

Nesting

Reference	Cardinality	Parameter
\IfcObjectDefinition.IsNestedBy		
\IfcObjectDefinition.IsNestedBy\IfcRelNests		
\IfcObjectDefinition.IsNestedBy\IfcRelNests.RelatedObjects		Type
\IfcObjectDefinition.IsNestedBy\IfcRelNests.RelatedObjects\IfcObject		

Process Type Assignment

Reference	Cardinality	Parameter
\IfcTypeProcess.OperatesOn		
\IfcTypeProcess.OperatesOn\IfcRelAssignsToProcess		
\IfcTypeProcess.OperatesOn\IfcRelAssignsToProcess.RelatedObjects		Type
\IfcTypeProcess.OperatesOn\IfcRelAssignsToProcess.RelatedObjects\IfcTypeResource		

Calendar

Reference	Cardinality	Parameter
\IfcWorkCalendar.WorkingTimes		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.RecurrenceType		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.RecurrenceType\IfcRecurrenceTypeEnum		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.TimePeriods		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.TimePeriods\IfcTimePeriod		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.StartTime		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.StartTime\IfcTime		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.EndTime		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.RecurrencePattern\IfcRecurrencePattern.TimePeriods\IfcTimePeriod.EndTime\IfcTime		

Reference	Cardinality	Parameter
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.Start		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.Start\IfcDate		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.Finish		
\IfcWorkCalendar.WorkingTimes\IfcWorkTime.Finish\IfcDate		

Aggregation

Reference	Cardinality	Parameter
\IfcObjectDefinition.PredefinedType		PredefinedType
\IfcObjectDefinition.IsDecomposedBy		
\IfcObjectDefinition.IsDecomposedBy\IfcRelAggregates		
\IfcObjectDefinition.IsDecomposedBy\IfcRelAggregates.RelatedObjects		RelatedObjects
\IfcObjectDefinition.IsDecomposedBy\IfcRelAggregates.RelatedObjects\IfcObject		

Classification

Reference	Cardinality	Parameter
\IfcObjectDefinition.HasAssociations		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.Identification		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.Identification\IfcIdentifier		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.Source		Source
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.Source\IfcLabel		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.Name		Name
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.Name\IfcLabel		
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.ReferenceTokens		Tokens
\IfcObjectDefinition.HasAssociations\IfcRelAssociatesClassification.RelatingClassification\IfcClassificationReference.ReferencedSource\IfcClassification.ReferenceTokens\IfcIdentifier		

Building Location

Reference	Cardinality	Parameter
-----------	-------------	-----------

\IfcBuilding.ElevationOfRefHeight		
\IfcBuilding.ElevationOfRefHeight\IfcLengthMeasure		
\IfcBuilding.ElevationOfTerrain		
\IfcBuilding.ElevationOfTerrain\IfcLengthMeasure		
\IfcBuilding.BuildingAddress		
\IfcBuilding.BuildingAddress\IfcPostalAddress		
\IfcBuilding.BuildingAddress\IfcPostalAddress.AddressLines		
\IfcBuilding.BuildingAddress\IfcPostalAddress.AddressLines\IfcLabel		
\IfcBuilding.BuildingAddress\IfcPostalAddress.Town		
\IfcBuilding.BuildingAddress\IfcPostalAddress.Town\IfcLabel		
\IfcBuilding.BuildingAddress\IfcPostalAddress.Region		
\IfcBuilding.BuildingAddress\IfcPostalAddress.Region\IfcLabel		
\IfcBuilding.BuildingAddress\IfcPostalAddress.PostalCode		
\IfcBuilding.BuildingAddress\IfcPostalAddress.PostalCode\IfcLabel		
\IfcBuilding.BuildingAddress\IfcPostalAddress.Country		
\IfcBuilding.BuildingAddress\IfcPostalAddress.Country\IfcLabel		

Placement

Reference	Cardinality	Parameter
\IfcProduct.ObjectPlacement	[0:1]	Type
\IfcProduct.ObjectPlacement\IfcLocalPlacement		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcC artesianPoint		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcC artesianPoint.Dim		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcC artesianPoint.Dim\IfcDimensionCount		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcC artesianPoint.Dim\IfcDimensionCount.Value=3		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirect ion		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirect ion.Dim		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirect		

Reference	Cardinality	Parameter
ion.Dim\IfcDimensionCount		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirection.Dim\IfcDimensionCount.Value=3		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection.Dim		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection.Dim\IfcDimensionCount		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection.Dim\IfcDimensionCount.Value=3		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.PlacementRelTo		
\IfcProduct.ObjectPlacement\IfcLocalPlacement.PlacementRelTo\IfcLocalPlacement		

Element Decomposition

Reference	Cardinality	Parameter
\IfcElement.IsDecomposedBy		
\IfcElement.IsDecomposedBy\IfcRelAggregates		
\IfcElement.IsDecomposedBy\IfcRelAggregates.RelatedObjects		RelatedObjects
\IfcElement.IsDecomposedBy\IfcRelAggregates.RelatedObjects\IfcElement		

Object Typing

Reference	Cardinality	Parameter
\IfcObject.IsTypedBy		
\IfcObject.IsTypedBy\IfcRelDefinesByType		
\IfcObject.IsTypedBy\IfcRelDefinesByType.RelatingType		Type
\IfcObject.IsTypedBy\IfcRelDefinesByType.RelatingType\IfcTypeObject		

Footprint Geometry

Reference	Cardinality	Parameter
\IfcProduct.Representation		
\IfcProduct.Representation\IfcProductDefinitionShape		
\IfcProduct.Representation\IfcProductDefinitionShape.Representations		
\IfcProduct.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation		
\IfcProduct.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier		
\IfcProduct.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel		
\IfcProduct.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel.Value=Footprint		

Reference	Cardinality	Parameter
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationType		RepresentationType
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationType\\IfcLabel		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationType\\IfcLabel.Value=GeometricCurveSet		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items		Geometry
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.Elements		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.Elements\\IfcBoundedCurve		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.Elements\\IfcBoundedCurve.StyledByItem	[0: 1]	
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.Elements\\IfcBoundedCurve.StyledByItem\\IfcStyledItem		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.Elements\\IfcBoundedCurve.StyledByItem\\IfcStyledItem.Styles		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.Elements\\IfcBoundedCurve.StyledByItem\\IfcStyledItem.Styles\\IfcCurveStyle		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.StyledByItem	[0: 1]	
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.StyledByItem\\IfcStyledItem		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.StyledByItem\\IfcStyledItem.Styles		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcGeometricCurveSet.StyledByItem\\IfcStyledItem.Styles\\IfcFillAreaStyle		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.ContextOfItems		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.ContextOfItems\\IfcGeometricRepresentationContext		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.LayerAssignments		
\\IfcProduct.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.LayerAssignments\\IfcPresentationLayerAssignment		

Body Geometry

Reference	Cardinality	Parameter
\\IfcElement.Representation		
\\IfcElement.Representation\\IfcProductDefinitionShape		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.ContextOfItems		

extOfItems		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.ContextOfItems\\IfcGeometricRepresentationContext		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationIdentifier		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationIdentifier\\IfcLabel		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationIdentifier\\IfcLabel.Value=Body		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationType		RepresentationType
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.RepresentationType\\IfcLabel		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items		Geometry
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcSolidModel		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcSolidModel.StyledByItem	[0:1]	
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcSolidModel.StyledByItem\\IfcStyledItem		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcSolidModel.StyledByItem\\IfcStyledItem.Styles		
\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items\\IfcSolidModel.StyledByItem\\IfcStyledItem.Styles\\IfcSurfaceStyle		

Spatial Containment

Reference	Cardinality	Parameter
\\IfcElement.ContainedInStructure		
\\IfcElement.ContainedInStructure\\IfcRelContainedInSpatialStructure		
\\IfcElement.ContainedInStructure\\IfcRelContainedInSpatialStructure.RelatingStructure		Structure
\\IfcElement.ContainedInStructure\\IfcRelContainedInSpatialStructure.RelatingStructure\\IfcSpatialStructureElement		

Grid

Reference	Cardinality	Parameter
\\IfcGrid.PredefinedType		GridType
\\IfcGrid.PredefinedType\\IfcGridTypeEnum		
\\IfcGrid.UAxes		
\\IfcGrid.UAxes\\IfcGridAxis		
\\IfcGrid.UAxes\\IfcGridAxis.AxisTag		
\\IfcGrid.UAxes\\IfcGridAxis.AxisTag\\IfcLabel		
\\IfcGrid.UAxes\\IfcGridAxis.AxisCurve		UCurve
\\IfcGrid.UAxes\\IfcGridAxis.AxisCurve\\IfcCurve		
\\IfcGrid.VAxes		

Reference	Cardinality	Parameter
\IfcGrid.VAxes\IfcGridAxis		
\IfcGrid.VAxes\IfcGridAxis.AxisTag		
\IfcGrid.VAxes\IfcGridAxis.AxisTag\IfcLabel		
\IfcGrid.VAxes\IfcGridAxis.AxisCurve		VCurve
\IfcGrid.VAxes\IfcGridAxis.AxisCurve\IfcCurve		

Site Location

Reference	Cardinality	Parameter
\IfcSite.RefLatitude		
\IfcSite.RefLatitude\IfcCompoundPlaneAngleMeasure		
\IfcSite.RefLongitude		
\IfcSite.RefLongitude\IfcCompoundPlaneAngleMeasure		
\IfcSite.RefElevation		
\IfcSite.RefElevation\IfcLengthMeasure		
\IfcSite.LandTitleNumber		
\IfcSite.LandTitleNumber\IfcLabel		
\IfcSite.SiteAddress		
\IfcSite.SiteAddress\IfcPostalAddress		
\IfcSite.SiteAddress\IfcPostalAddress.AddressLines		
\IfcSite.SiteAddress\IfcPostalAddress.AddressLines\IfcLabel		
\IfcSite.SiteAddress\IfcPostalAddress.Town		
\IfcSite.SiteAddress\IfcPostalAddress.Town\IfcLabel		
\IfcSite.SiteAddress\IfcPostalAddress.Region		
\IfcSite.SiteAddress\IfcPostalAddress.Region\IfcLabel		
\IfcSite.SiteAddress\IfcPostalAddress.Country		
\IfcSite.SiteAddress\IfcPostalAddress.Country\IfcLabel		
\IfcSite.SiteAddress\IfcPostalAddress.PostalCode		
\IfcSite.SiteAddress\IfcPostalAddress.PostalCode\IfcLabel		

Space Coverings

Reference	Cardinality	Parameter
\IfcSpace.HasCoverings		
\IfcSpace.HasCoverings\IfcRelCoversSpaces		
\IfcSpace.HasCoverings\IfcRelCoversSpaces.RelatedCoverings		
\IfcSpace.HasCoverings\IfcRelCoversSpaces.RelatedCoverings\IfcCovering		

Reference	Cardinality	Parameter
\IfcSpace.HasCoverings\IfcRelCoversSpaces.RelatedCoverings\IfcCovering.PredefinedType		CoveringType
\IfcSpace.HasCoverings\IfcRelCoversSpaces.RelatedCoverings\IfcCovering.PredefinedType\IfcCoveringTypeEnum		

Properties for Types

Reference	Cardinality	Parameter
\IfcTypeObject.HasPropertySets		
\IfcTypeObject.HasPropertySets\IfcPropertySet		
\IfcTypeObject.HasPropertySets\IfcPropertySet.Name		Name
\IfcTypeObject.HasPropertySets\IfcPropertySet.Name\IfcLabel		
\IfcTypeObject.HasPropertySets\IfcPropertySet.HasProperties		
\IfcTypeObject.HasPropertySets\IfcPropertySet.HasProperties\IfcProperty		

Ports

Reference	Cardinality	Parameter
\IfcDistributionElement.PredefinedType		PredefinedType
\IfcDistributionElement.IsNestedBy		
\IfcDistributionElement.IsNestedBy\IfcRelNests		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.Name		Name
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.Name\IfcLabel		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.FlowDirection		Flow
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.FlowDirection\IfcFlowDirectionEnum		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.PredefinedType		Type
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.PredefinedType\IfcDistributionSystemEnum		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcCartesianPoint		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis		

Reference	Cardinality	Parameter
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet.MaterialProfiles		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.ForProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement		

Reference	Cardinality	Parameter
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcCartesianPoint		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.Name		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.Name\IfcLabel		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.FlowDirection		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.FlowDirection\IfcFlowDirectionEnum		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.PredefinedType		
\IfcDistributionElement.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.IsDeclaredBy\IfcRelDefinesByObject.RelatingObject\IfcDistributionPort.PredefinedType\IfcDistributionSystemEnum		

Product Type Assignment

Reference	Cardinality	Parameter
\IfcTypeProduct.ReferencedBy		
\IfcTypeProduct.ReferencedBy\IfcRelAssignsToProduct		
\IfcTypeProduct.ReferencedBy\IfcRelAssignsToProduct.RelatedObjects		Type
\IfcTypeProduct.ReferencedBy\IfcRelAssignsToProduct.RelatedObjects\IfcTypeProcess		

Type-Based Ports

Reference	Cardinality	Parameter
\IfcDistributionElementType.IsNestedBy		
\IfcDistributionElementType.IsNestedBy\IfcRelNests		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects		

Reference	Cardinality	Parameter
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.Name		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.Name\IfcLabel		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.FlowDirection		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.FlowDirection\IfcFlowDirectionEnum		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.PredefinedType		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.PredefinedType\IfcDistributionSystemEnum		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Location\IfcCartesianPoint		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.Axis\IfcDirection		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection		
\IfcDistributionElementType.IsNestedBy\IfcRelNests.RelatedObjects\IfcDistributionPort.ObjectPlacement\IfcLocalPlacement.RelativePlacement\IfcAxis2Placement3D.RefDirection\IfcDirection		

Surface Geometry

Reference	Cardinality	Parameter
\IfcElement.Representation		
\IfcElement.Representation\IfcProductDefinitionShape		
\IfcElement.Representation\IfcProductDefinitionShape.Representations		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems\IfcGeometricRepresentationContext		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Re		

Reference	Cardinality	Parameter
presentationIdentifier\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel.Value=Surface		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType		Representation Type
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel.Value=Surface3D		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items		Geometry
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcBoundedSurface		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcBoundedSurface.StyledByItem	[0:1]	
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcBoundedSurface.StyledByItem\IfcStyledItem		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcBoundedSurface.StyledByItem\IfcStyledItem.Styles		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcBoundedSurface.StyledByItem\IfcStyledItem.Styles\IfcSurfaceStyle		

Axis Geometry

Reference	Cardinality	Parameter
\IfcElement.Representation		
\IfcElement.Representation\IfcProductDefinitionShape		
\IfcElement.Representation\IfcProductDefinitionShape.Representations		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems\IfcGeometricRepresentationContext		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel.Value=Axis		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType		Representation Type
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel.Value=Curve3D		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items		Geometry

Reference	Cardinality	Parameter
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.It ems\IfcBoundedCurve		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.It ems\IfcBoundedCurve.StyledByItem	[0:1]	
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.It ems\IfcBoundedCurve.StyledByItem\IfcStyledItem		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.It ems\IfcBoundedCurve.StyledByItem\IfcStyledItem.Styles		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.It ems\IfcBoundedCurve.StyledByItem\IfcStyledItem.Styles\IfcCurveStyle		

Group Assignment

Reference	Cardinality	Parameter
\IfcGroup.IsGroupedBy		
\IfcGroup.IsGroupedBy\IfcRelAssignsToGroup		
\IfcGroup.IsGroupedBy\IfcRelAssignsToGroup.RelatedObjects		Type
\IfcGroup.IsGroupedBy\IfcRelAssignsToGroup.RelatedObjects\IfcProduct		

Material Profile Set Usage

Reference	Cardinality	Parameter
\IfcProduct.HasAssociations		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Name		Name
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile		
\IfcProduct.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSetUsage.F orProfileSet\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef		

Port Connectivity

Reference	Cardinality	Parameter
-----------	-------------	-----------

Reference	Cardinality	Parameter
\IfcDistributionPort.ConnectedTo		
\IfcDistributionPort.ConnectedTo\IfcRelConnectsPorts		
\IfcDistributionPort.ConnectedTo\IfcRelConnectsPorts.RelatedPort		
\IfcDistributionPort.ConnectedTo\IfcRelConnectsPorts.RelatedPort\IfcDistributionPort		
\IfcDistributionPort.ConnectedTo\IfcRelConnectsPorts.RealizingElement		
\IfcDistributionPort.ConnectedTo\IfcRelConnectsPorts.RealizingElement\IfcFlowSegment		

Cost

Reference	Cardinality	Parameter
\IfcCostItem.CostValues		
\IfcCostItem.CostValues\IfcCostValue		
\IfcCostItem.CostValues\IfcCostValue.AppliedValue		
\IfcCostItem.CostValues\IfcCostValue.AppliedValue\IfcMonetaryMeasure		
\IfcCostItem.CostValues\IfcCostValue.ApplicableDate		
\IfcCostItem.CostValues\IfcCostValue.ApplicableDate\IfcDate		
\IfcCostItem.CostValues\IfcCostValue.FixedUntilDate		
\IfcCostItem.CostValues\IfcCostValue.FixedUntilDate\IfcDate		

Resource Cost

Reference	Cardinality	Parameter
\IfcConstructionResource.BaseCosts		CostType
\IfcConstructionResource.BaseCosts\IfcAppliedValue		
\IfcConstructionResource.BaseCosts\IfcAppliedValue.Name		CostName
\IfcConstructionResource.BaseCosts\IfcAppliedValue.Name\IfcLabel		
\IfcConstructionResource.BaseCosts\IfcAppliedValue.AppliedValue		ValueType

Resource Quantity

Reference	Cardinality	Parameter
\IfcConstructionResource.BaseQuantity		QuantityType
\IfcConstructionResource.BaseQuantity\IfcPhysicalSimpleQuantity		
\IfcConstructionResource.BaseQuantity\IfcPhysicalSimpleQuantity.Name		QuantityName
\IfcConstructionResource.BaseQuantity\IfcPhysicalSimpleQuantity.Name\IfcLabel		

Resource Assignment

Reference	Cardinality	Parameter
\IfcResource.ResourceOf		
\IfcResource.ResourceOf\IfcRelAssignsToResource		

Reference	Cardinality	Parameter
\IfcResource.ResourceOf\IfcRelAssignsToResource.RelatedObjects		Type
\IfcResource.ResourceOf\IfcRelAssignsToResource.RelatedObjects\IfcActor		
\IfcResource.ResourceOf\IfcRelAssignsToResource.RelatedObjects\IfcProduct		

Resource Cost Rate

Reference	Cardinality	Parameter
\IfcConstructionResourceType.BaseCosts		
\IfcConstructionResourceType.BaseCosts\IfcCostValue		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.AppliedValue		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.AppliedValue\IfcMonetaryMeasure		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.UnitBasis		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.UnitBasis\IfcMeasureWithUnit		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.UnitBasis\IfcMeasureWithUnit.ValueComponent		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.UnitBasis\IfcMeasureWithUnit.ValueComponent\IfcValue		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.UnitBasis\IfcMeasureWithUnit.UnitComponent		
\IfcConstructionResourceType.BaseCosts\IfcCostValue.UnitBasis\IfcMeasureWithUnit.UnitComponent\IfcUnit		

Interference

Reference	Cardinality	Parameter
\IfcElement.InterferesElements		
\IfcElement.InterferesElements\IfcRelInterferesElements		
\IfcElement.InterferesElements\IfcRelInterferesElements.RelatingElement		
\IfcElement.InterferesElements\IfcRelInterferesElements.RelatingElement\IfcElement		

Material Profile Set

Reference	Cardinality	Parameter
\IfcElementType.HasAssociations		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material	[0: 1]	
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial		

Reference	Cardinality	Parameter
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation.Representations		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation.Representations\IfcStyledRepresentation		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation.Representations\IfcStyledRepresentation.Items		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation.Representations\IfcStyledRepresentation.Items\IfcStyledItem		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation.Representations\IfcStyledRepresentation.Items\IfcStyledItem.Styles		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasRepresentation\IfcMaterialDefinitionRepresentation.Representations\IfcStyledRepresentation.Items\IfcStyledItem.Styles\IfcPresentationStyle		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasProperties		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasProperties\IfcMaterialProperties		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasProperties\IfcMaterialProperties.Properties		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Material\IfcMaterial.HasProperties\IfcMaterialProperties.Properties\IfcProperty		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile	[1:1]	
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef.HasProperties		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef.HasProperties\IfcProfileProperties		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef.HasProperties\IfcProfileProperties.Properties		
\IfcElementType.HasAssociations\IfcRelAssociatesMaterial.RelatingMaterial\IfcMaterialProfileSet.MaterialProfiles\IfcMaterialProfile.Profile\IfcProfileDef.HasProperties\IfcProfileProperties.Properties\IfcProperty		

Clearance Geometry

Reference	Cardinality	Parameter
\IfcElement.Representation		
\IfcElement.Representation\IfcProductDefinitionShape		
\IfcElement.Representation\IfcProductDefinitionShape.Representations		

Reference	Cardinality	Parameter
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems\IfcGeometricRepresentationContext		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel.Value=Clearance		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType		RepresentationType
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items		Geometry
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcBoundedSurface		

Lighting Geometry

Reference	Cardinality	Parameter
\IfcElement.Representation		
\IfcElement.Representation\IfcProductDefinitionShape		
\IfcElement.Representation\IfcProductDefinitionShape.Representations		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.ContextOfItems\IfcGeometricRepresentationContext		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationIdentifier\IfcLabel.Value=Lighting		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType		Type
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.RepresentationType\IfcLabel.Value=LightSource		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items		Geometry
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcLightSource		
\IfcElement.Representation\IfcProductDefinitionShape.Representations\IfcShapeRepresentation.Items\IfcLightSource.LightColour		

\\IfcElement.Representation\\IfcProductDefinitionShape.Representations\\IfcShapeRepresentation.Items \\IfcLightSource.LightColour\\IfcColourRgb		
--	--	--

4.7.6.3.5 Concept business rule description

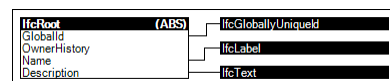
Each concept template is described in a subsection as follows, with diagrams indicating usage of attributes and entities reflecting defined business rules.

Identity

An object needs to be identifiable for accurate processing by both human and automated processes. Identification may be through several attributes such as Identification, Name, or GUID. The GUID is compressed for the purpose of being exchanged within an IFC data set - the compressed GUID is referred to as "IFC-GUID". While the IFC-GUID is normally generated automatically and has to be persistent, the Identification may relate to other informal registers but should be unique within the set of objects of the same type. The Name and Description should allow any object to be identified in the context of the project or facility being modelled.

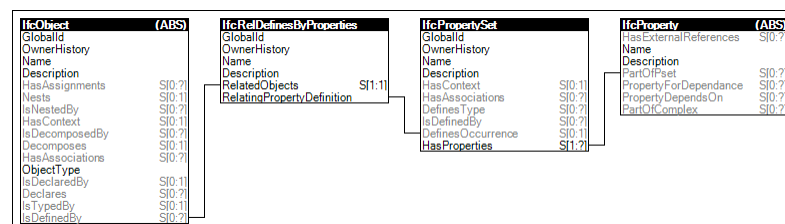
Various objects may have additional identifications that may be human-readable and/or may be structured through classification association.

Various file formats may use additional identifications of instances for serialization purposes, however there is no requirement or guarantee for such identifications to remain the same between revisions or across applications. For example, the IFC-SPF file format lists each instance with a 64-bit integer that is unique within the particular file.



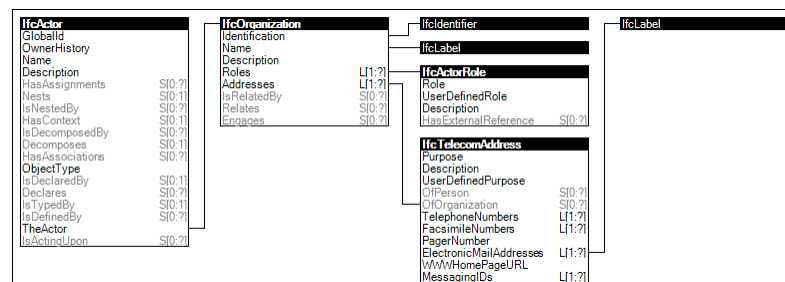
Properties for Occurrences

Any specialization of *object* can be related to multiple *property set occurrences*. A property set contains multiple *property occurrences*. The data types of property occurrences are single value, enumerated value, bounded value, table value, reference value, list value, and combination of property occurrences.



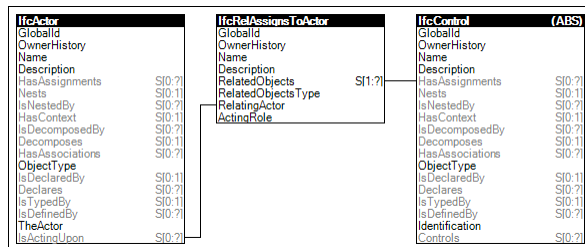
Organization Role

Contact information indicates roles and addresses of people and organizations.



Actor Assignment

Actors may have assignments indicating objects for which they have responsibility. An example of such assignment is a work order assigned to an organization.



Revision Control

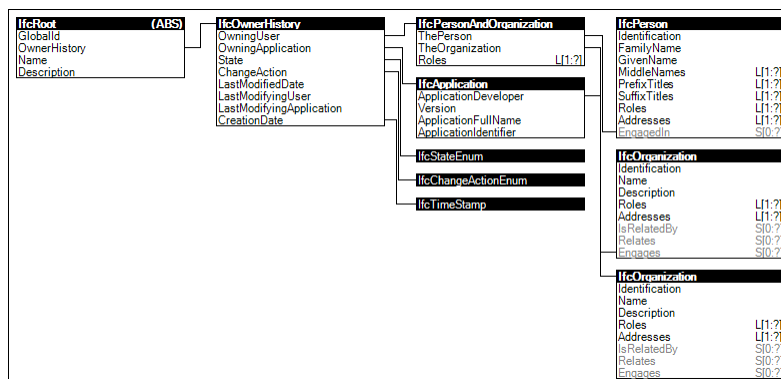
While objects may reflect a final state, they may also be continually revised over the course of a project lifecycle and reflect transient state. For scenarios of multiple users making updates to the same information, there is a concept of local copies of information based upon a shared repository supporting multiple users. Such shared repository is often referred to as a *model server*. A model server is similar in concept to a document revision server, but is able to identify changes declared on a per-object basis rather than inferring changes from differences in text. A model server has a concept of revisions on a per-project basis, where each revision consists of a set of changes to contained objects by a particular user at a particular time.

To support a model server scenario, each object may be marked with a *change action* indicating the object was added, modified, deleted, or has no change since the project was retrieved from the server at a particular revision sequence. Given an object's identity (IFC-GUID) and change action, the state of the object may be merged when submitted to a model server. An object is considered modified when any of its *direct attributes* change, attributes on a referenced resource definition (any entity not deriving from *IfcRoot*) change, items are added or removed from sets, or items are added, removed, or reordered within lists.

For cases when multiple users make conflicting changes to the same objects, users may choose to keep their own changes, accept changes from others, merge both changes, or a combination thereof upon submitting to a server. Alternatively, to avoid such merge scenario and coordinate work, objects may be locked such that a particular user has exclusive access to read and/or write a particular object at the current time.

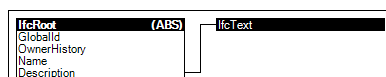
Project libraries may also be retrieved from model servers having particular revision, and potentially different server URI than the referencing Project. As a project may include multiple revisions of the same project library (a common scenario when multiple users are involved using libraries revised by others), the *IfcRoot.ObjectIdentifier* IFC-GUID is only valid within the scope of the referencing project, and a separate *library reference* identifies a project library based object within its originating model server.

Finally, objects may also carry informational attributes indicating when an object was created, who, when, and what application was used to last modify an object, and who currently owns the object, potentially having exclusive use according to its lock state.



Descriptions

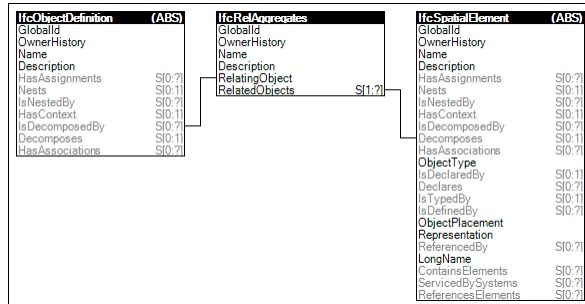
Objects may have descriptions included to aid in human identification of the object.



Spatial Decomposition

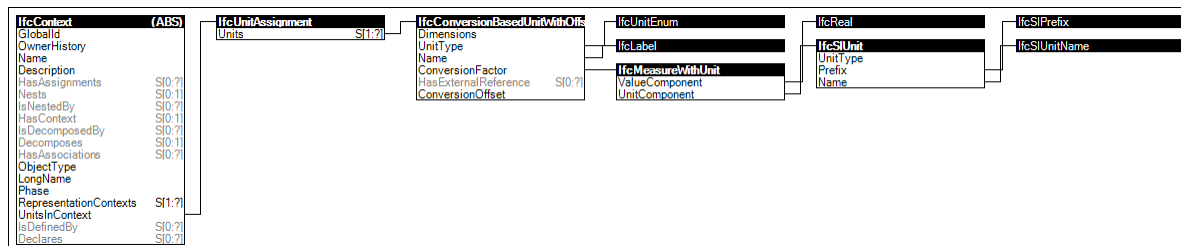
Provision of a spatial structure of the project by aggregating spatial structure elements. The spatial structure is a hierarchical tree of spatial structure elements (site, building, storey, space) ultimately assigned to the project. Decomposition refers to the relationship to lower level elements (e.g. this storey has spaces).

The order of spatial structure elements being included in the concept are from high to low level: *IfcProject*, *IfcSite*, *IfcBuilding*, *IfcBuildingStorey*, *IfcSpace*. Therefore an spatial structure element can only has parts of an element at the same or lower level.



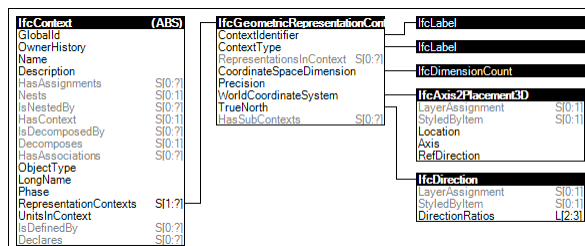
Conversion Units

Conversion units are defined according to a conversion factor (and conversion offset for temperature) relative to a specified base SI unit.



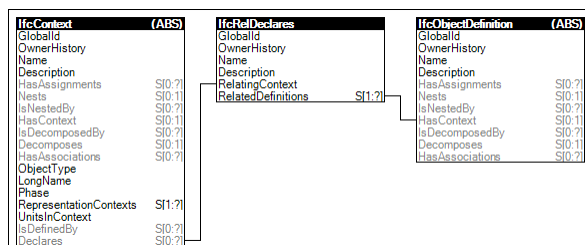
Project Context

A project representation context indicates the coordinate system orientation, direction of true north, precision, and other values that apply to all geometry within a project or project library.



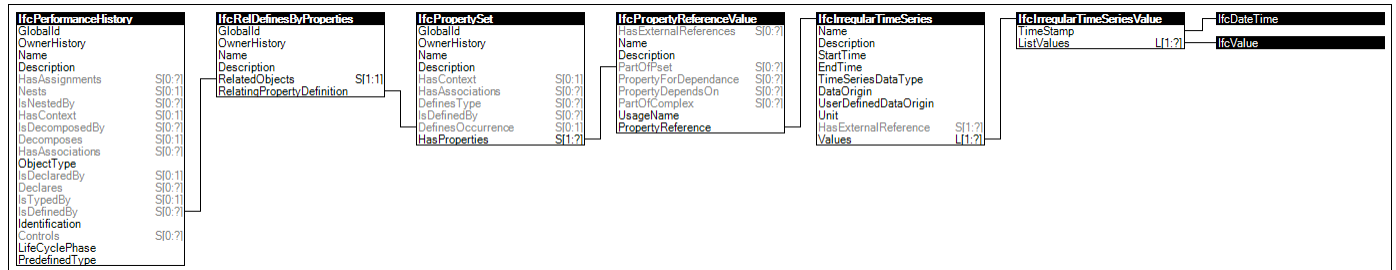
Project Declaration

The project provides a directory of objects contained within using declaration relationships.



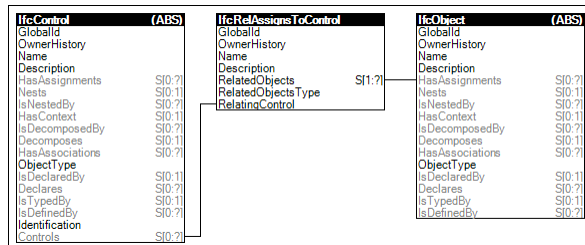
Properties for Performance

For performance history, properties are in the form of time series, for tracking data at points in time.



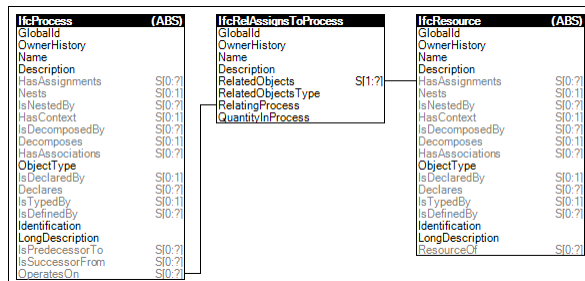
Control Assignment

Controls may have assignments indicating objects that must observe the established requirements. An example of such assignment is a labor resource assigned to a calendar.



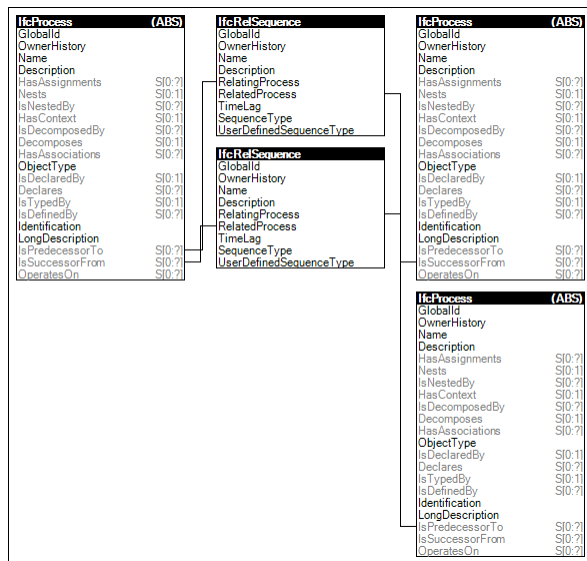
Process Assignment

Processes may have assignments indicating resources consumed or occupied by the process. An example of such assignment is a carpenter labor resource building a wall.



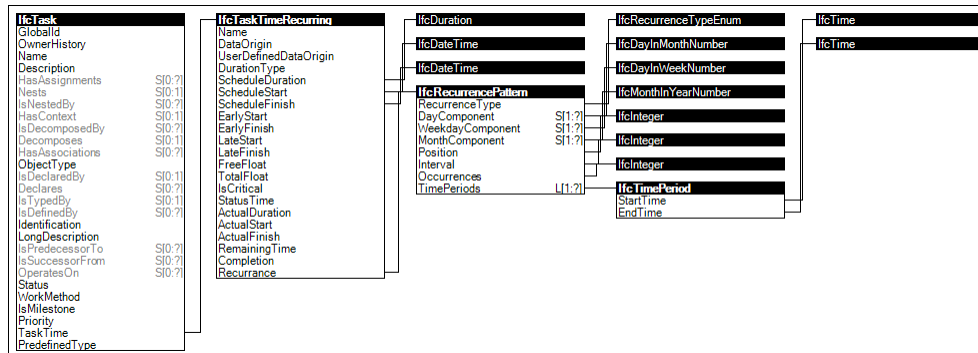
Sequential Connectivity

Processes that occur in time use this relationship to indicate the order of occurrence, such as for tasks, procedures, and events.



Task Scheduling

Tasks may be scheduled to run continuously, at a single period in time, or multiple recurring periods in time.



Nesting

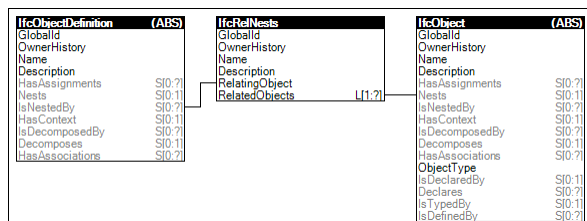
Nesting indicates an ordered arrangement relationship.

Nesting is used on building elements to indicate features placed in sequence such as ports.

Nesting is used on control objects to indicate specification hierarchies.

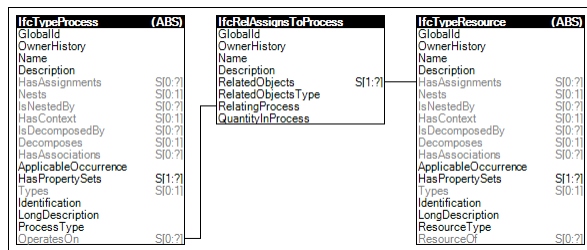
Nesting is used on process objects to indicate subordinate task details.

Nesting is used on resource objects to indicate subordinate resource allocations.



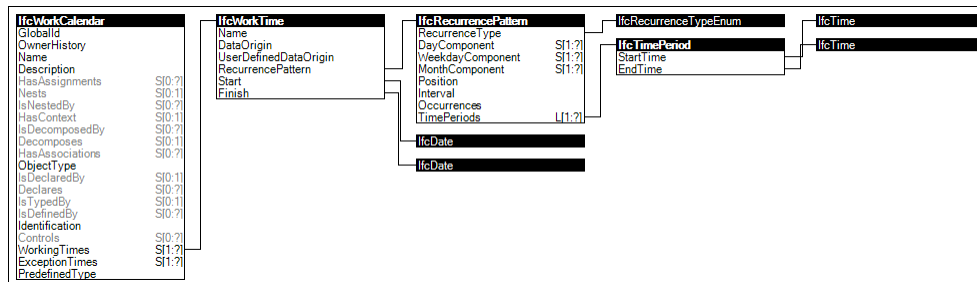
Process Type Assignment

Process types may have assignments indicating re-usable resource types for which occurrences may be consumed or occupied by occurrences of the process type. An example of such assignment is a concrete mixer resource type for delivering concrete.



Calendar

Calendar information is used to filter other objects to indicate time periods during which the control applies.

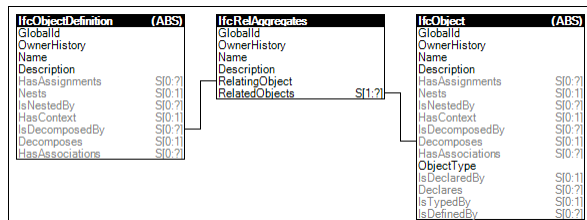


Aggregation

Aggregation indicates an unordered part composition relationship.

Aggregation is used on building elements to indicate parts such as studs within a wall.

Aggregation is used on systems to indicate subsystems such as branch circuits.

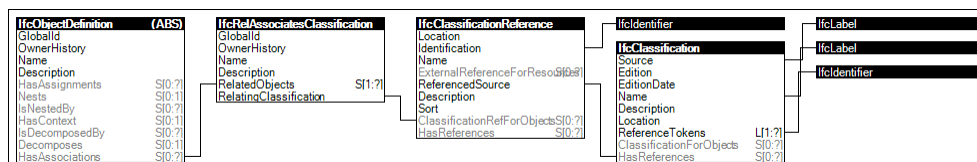


Classification

Objects, type objects, properties, and some resource schema entities can be further described by associating references to external sources of information. The source of information can be:

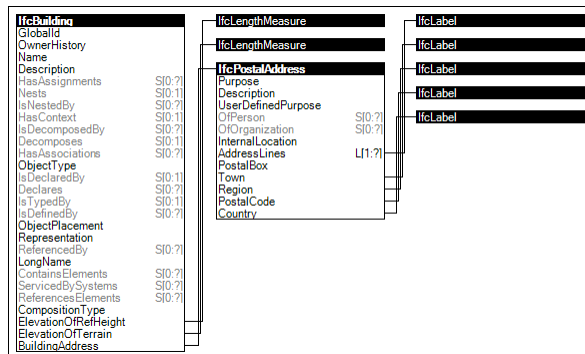
- a classification system;
- a dictionary server;
- any external catalogue that classifies the object further;
- a service that combine the above features.

An individual item within the external source of information can be selected. It then applies the inherent meaning of the item to the object or property.



Building Location

The building location may indicate the address as found on a map.



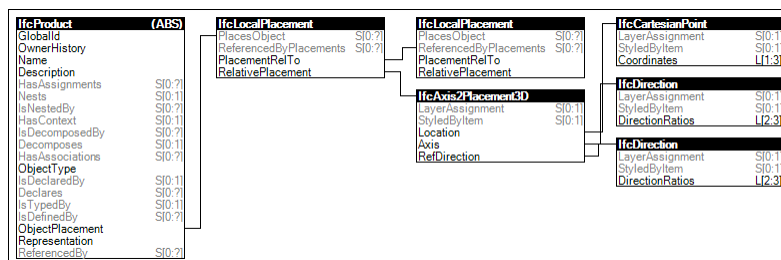
Placement

Product occurrences can be placed in 3D space relative to where they are contained. Placement is defined by a relative position (X, Y, Z coordinates), a horizontal reference direction, and a vertical axis direction. At the outermost level, relative directions are defined according to representation context; for example, +X may point east, +Y may point north, and +Z may point up.

Placement follows aggregation and containment relationships as follows:

- at the outermost level, a site is globally positioned according to latitude, longitude, and elevation;
- for spatial structures, positioning is relative to aggregation. For example, a site may aggregate multiple buildings, each building may aggregate multiple building storeys, and each building storey may aggregate multiple spaces;
- for building elements, positioning is relative to the containing spatial structure. For example, a building storey may contain slabs, walls, columns, and beams;
- for aggregated parts, positioning is relative to aggregation. For example, a staircase may aggregate one or more stair flights;
- for feature elements, positioning is relative to the affected building element. For example, an opening element is positioned relative to the wall it voids, which in turn is positioned relative to a building storey;
- for fillings, positioning is relative to the filled opening. For example, a door is positioned relative to an opening which in turn is positioned relative to a wall;
- for distribution ports, positioning is relative to the containing distribution element. For example, an air terminal may have a port connection for a duct segment or fitting;
- for distribution elements, positioning is relative to the containing spatial structure, however may be constrained by port connections. For example, a electrical junction box may fill an opening within a wall, and the junction box may contain ports for contained outlets or switches; the placement of such connected elements is constrained relative to connected port of the junction box. As another example, an air terminal may fill a ceiling covering which is placed relative to a space; the placement of a connecting duct fitting may be constrained relative to the air terminal.

If a containing spatial structure contains a grid, then placement may also be based relative to grid coordinates.



Element Decomposition

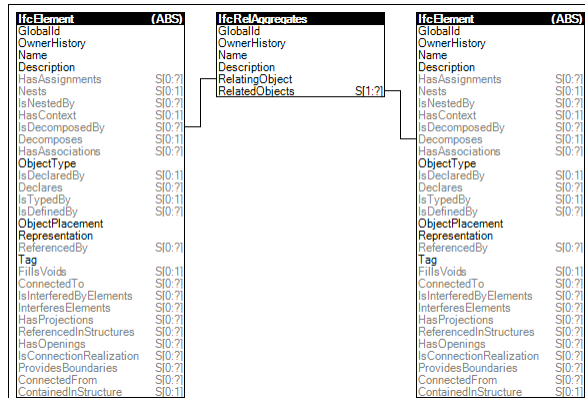
Provision of an aggregation structure where the element, representing the composite, is decomposed into parts represented by other elements.

The composite then provides, if such concepts are in scope of the Model View Definition, exclusively the following:

- **Placement** — the common object coordinate system to which the parts are placed relative

By default the following constraints apply to an element being decomposed by *Element Decomposition*:

- **Body Geometry** — composite is constructed from the sum of the *Body Geometry* of the parts;
- the composite shall not have an own *Body Geometry*, body geometry is provided at the parts;
- the composite shall not have an own *Material* assignment, material is assigned to the parts.



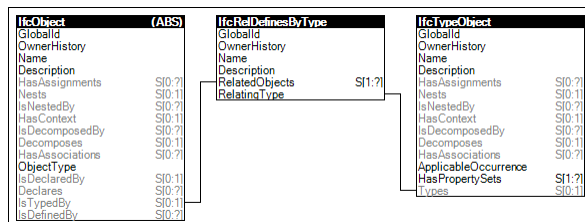
Object Typing

Object Occurrences may be defined by a particular *Object Type*, where such type describes common characteristics. Such characteristics include common properties, shapes, materials, composition, and other concepts described at particular entities. An object occurrence may have similar state as its object type, overridden state for particular characteristics, or have no defined type object.

A pair of *entities* are defined for various object occurrences and object types, where such object occurrence entity may only be defined using a particular object type entity. For example, the *IfcTank* occurrence object entity has a corresponding *IfcTankType* type object entity.

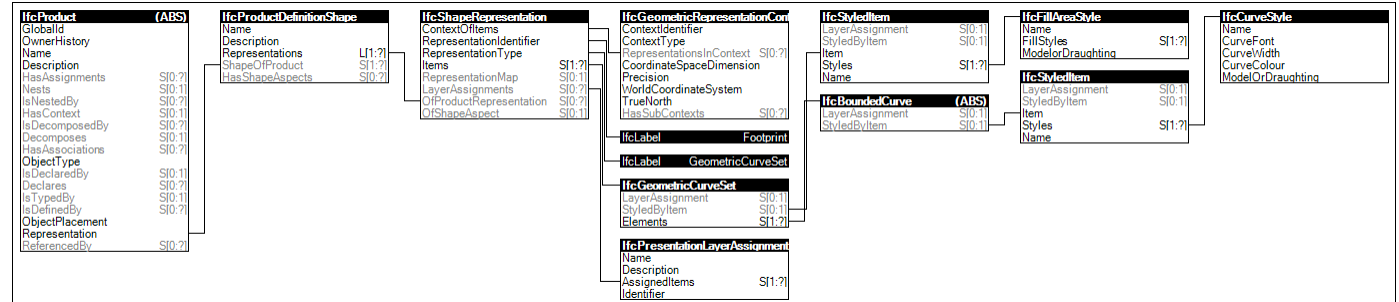
Many object occurrence and object type entities have an *attribute* named *PredefinedType* consisting of a specific *enumeration*. Such predefined type essentially provides another level of inheritance to further differentiate objects without the need for additional *entities*. Predefined types are not just informational; various rules apply such as applicable *property sets*, part composition, and distribution ports.

For scenarios of object types having part compositions, such parts may be reflected at object occurrences having separate state. For example, a *wall type* may define a particular arrangement of studs, a *wall occurrence* may reflect the same arrangement of studs, and studs within the wall occurrence may participate in specific relationships that do not exist at the type such as being connected to an electrical junction box.



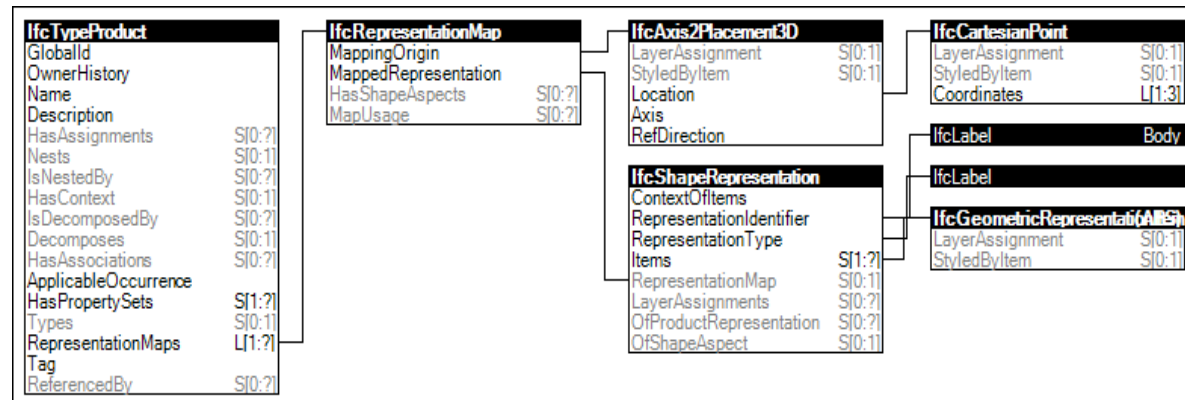
Footprint Geometry

Elements filling a boundary provide a 'Footprint' representation indicating a rectangle or any arbitrary set of outer and inner boundary curves. Examples of such elements include slabs and spaces. For elements that have a material layer set association indicating material thicknesses, a 'Body' representation may be generated based on the footprint and material layers. Fill area styles may indicate particular colors, tiles, or hatching for 2D rendering.



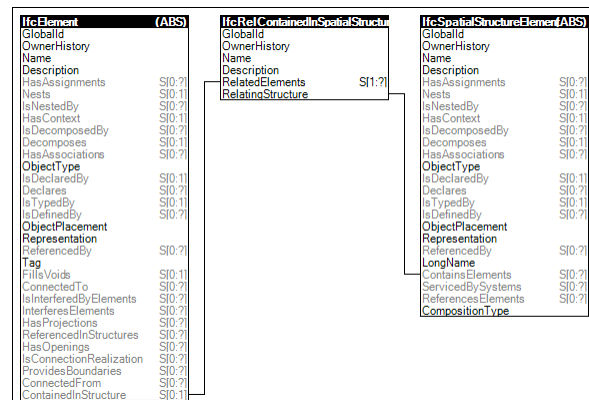
Body Geometry

Elements may have a 'Body' representation describing the volumetric shape of the object. Such representation may be used for 3D rendering or quantity take-off. Geometry may be based on boundary representations describing outer faces, primitives such as spheres or cones, swept solids such as profile extrusions or revolutions, Constructive Solid Geometry (CSG) such as clippings or subtractions of other shapes, or Non-Uniform Rational B-Spline (NURBS) geometry. Surface styles may indicate particular colors, textures, and reflectance for 3D rendering.



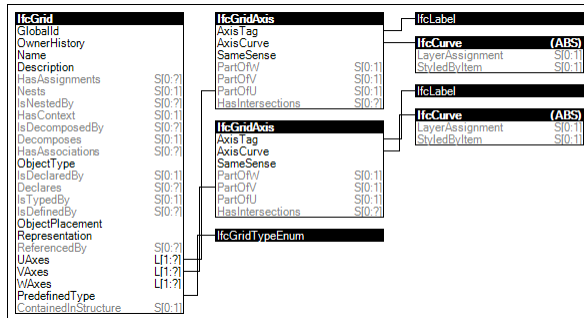
Spatial Containment

Spatial structures may contain physical elements, including building elements, distribution elements, and furnishing elements.



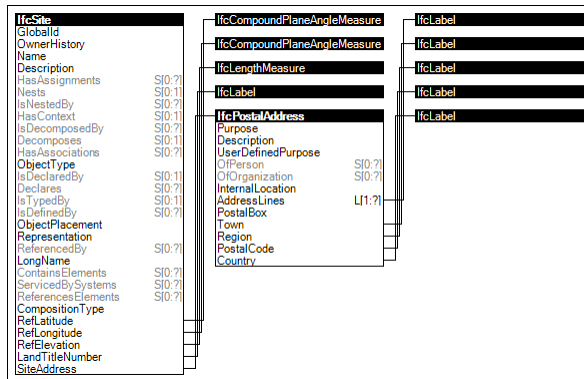
Grid

Grids are used for layout according to rectangular, triangular or circular patterns

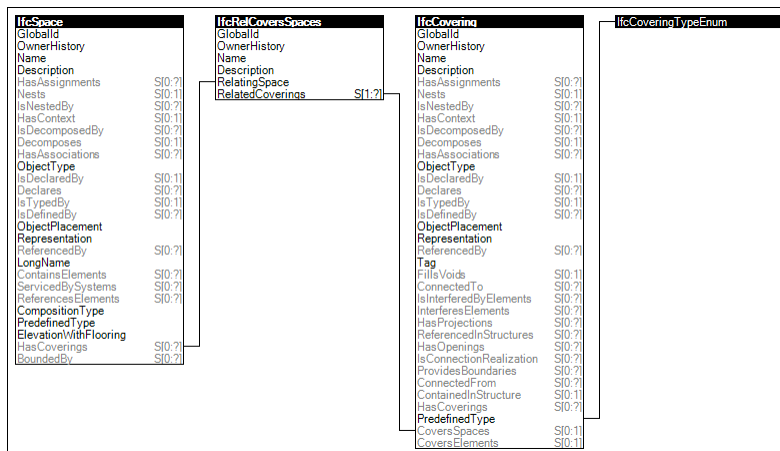


Site Location

The site location may be used to determine climate conditions and applicable building codes.

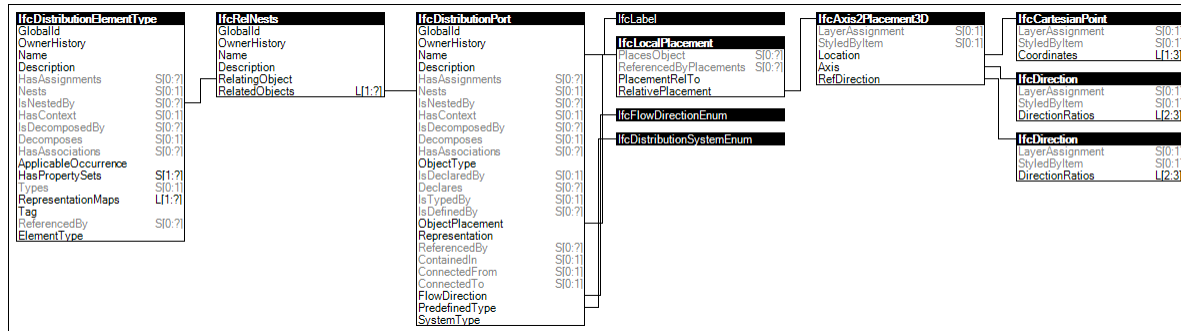


Space Coverings



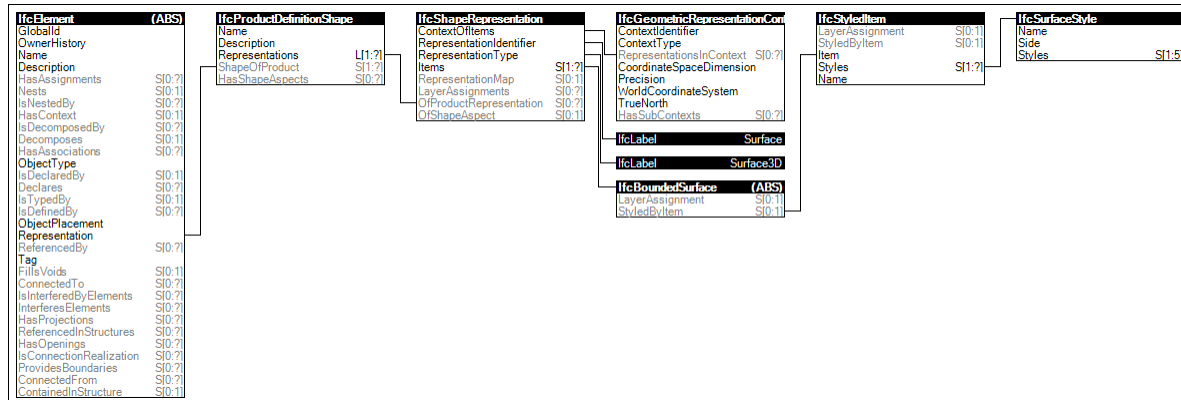
Type-Based Ports

Ports may be specified on types, following the same rules as defined for corresponding occurrences.



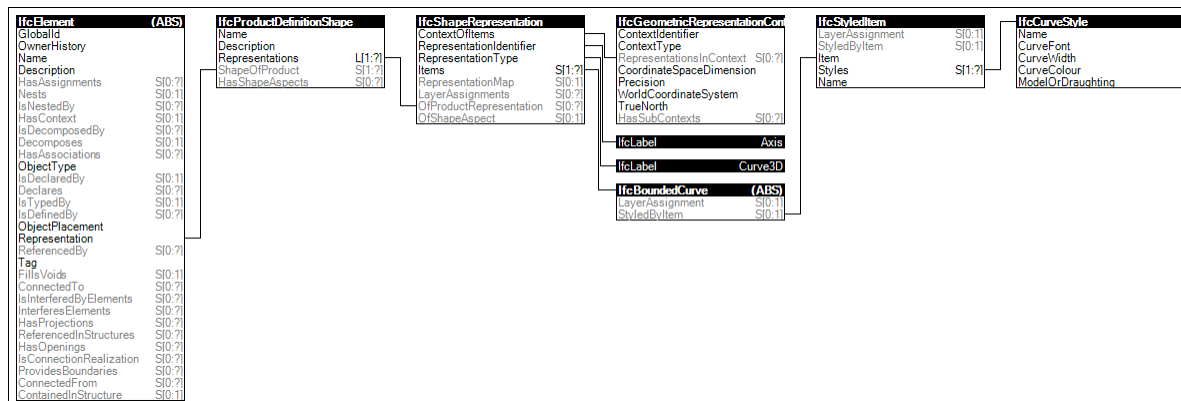
Surface Geometry

Elements may have a 'Surface' representation describing the outer surface of the object. Such representation may be used for hit-testing objects having part composition such as framed walls.



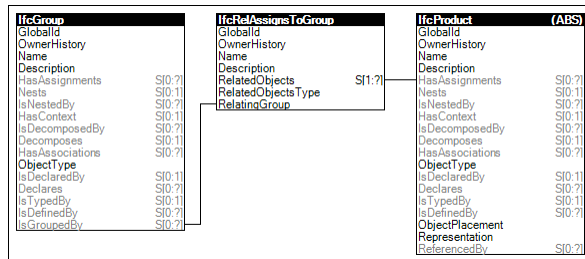
Axis Geometry

Elements following a path provide an 'Axis' representation indicating a line segment or any arbitrary open bounded curve. Examples of such elements include walls, beams, columns, pipes, ducts, and cables. For elements that have a material profile set association indicating cross-section, a 'Body' representation may be generated based on the axis curve and material profiles. Curve styles may indicate particular colors, line thicknesses, and dash patterns for 2D rendering.



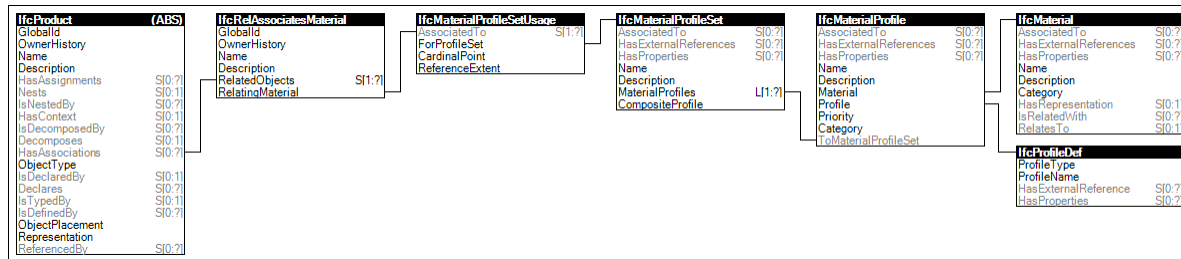
Group Assignment

Groups may have assignments indicating products that are members of the group. An example of such assignment is an air handler belonging to an air conditioning system.



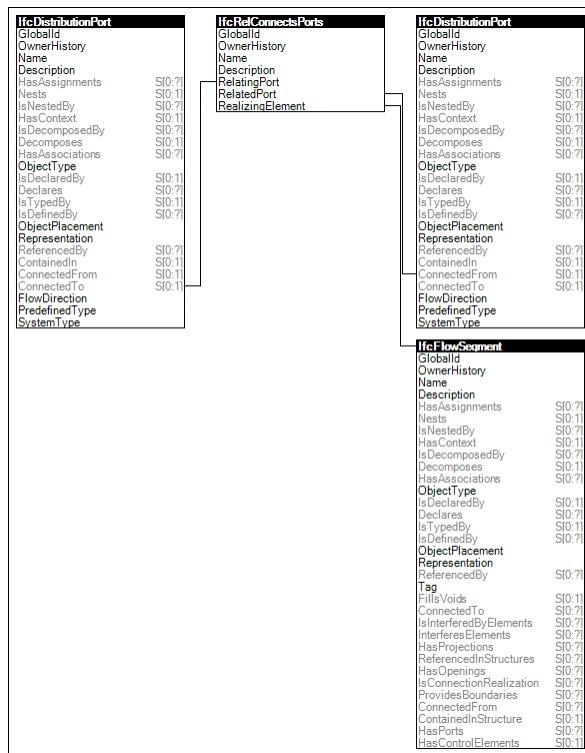
Material Profile Set Usage

Material profile set usage defines layout at occurrences to indicate the offset from the 'Axis' reference curve according to cardinal point, and a reference extent such as for a default column height.



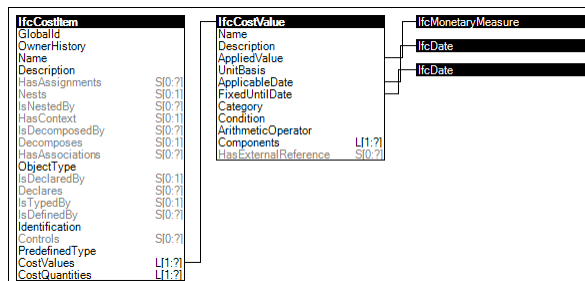
Port Connectivity

Ports on objects may be connected using elements such as cables, ducts, or pipes. Once Components within a System has some ports, then the connectivity should be complete and continuous. The presence of ports for air, water and electrical connections on complex equipment does not imply that all such connectivity is expected: only that if for example the HVAC segments and fittings have ports, then they will need to connect properly to the equipment's air ports.



Cost

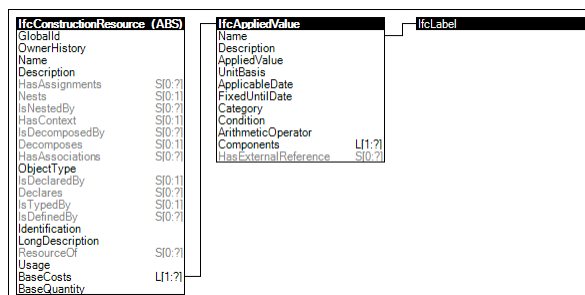
Cost information is used to indicate rate structures within a cost schedule which are applicable to assigned objects.



Resource Cost

Resources can have associated costs indicating financial costs and environmental impacts incurred according to a specified base quantity.

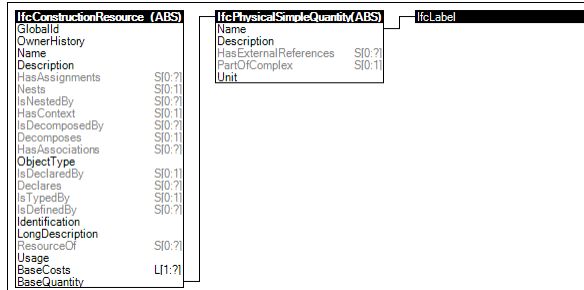
Each cost value may be defined using a constant amount or calculated according to specified formula.



Resource Quantity

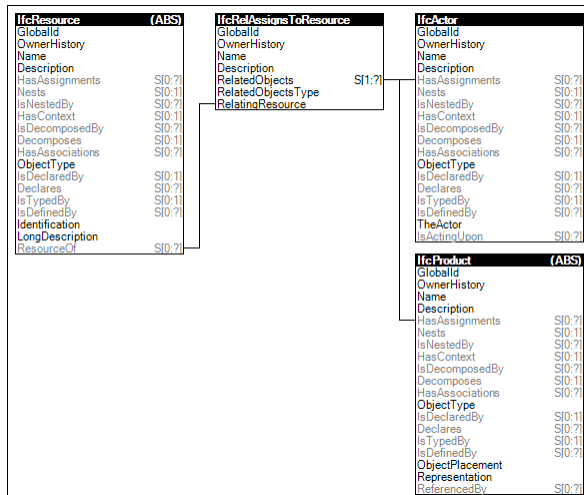
Resources may be defined according to a base quantity, where assigned tasks consume such amount of resource relative to an output quantity.

For work-based resources such as labor and equipment, quantities are based on time. For product-based resources, quantities are based on count. For material-based resources, quantities are based on volume.

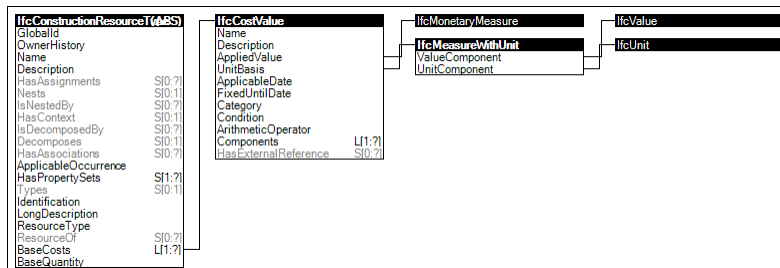


Resource Assignment

Resources may have assignments indicating sources available to be used. An example of such assignment is a person fulfilling a carpenter labor resource.



Resource Cost Rate

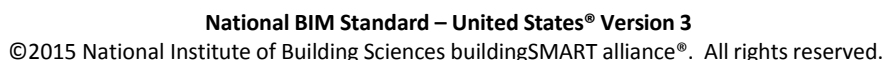


Elements may interfere with other elements, such as cable carriers going through walls. The interference relation enables precedence of interfering elements to be asserted.



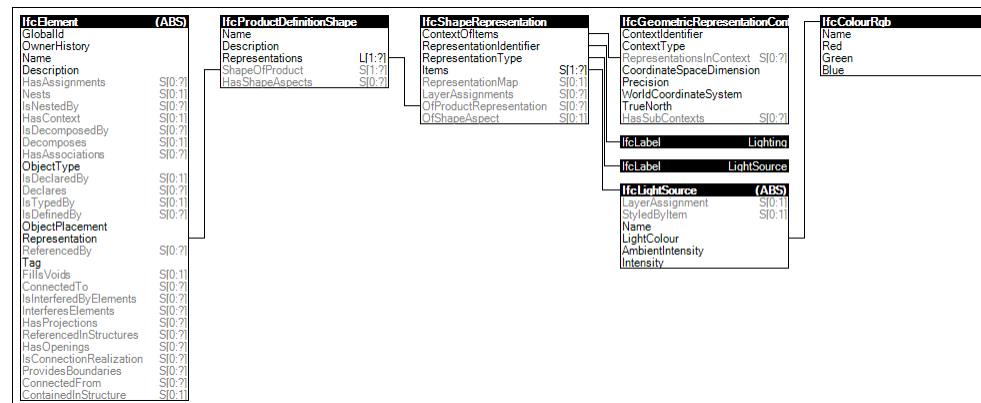
UtlItem Type (AIFS) UtlItem OwnerHistory Name Description HasAssignments HasItems HasInstability HasInstabilityReason HasDecomposability HasDecomposition HasAssociations ApplicableOccurrence HasPattern/Get Type RepresentationMaps Tag Decomposability Element Type	UtlItemAssociation Material UtlItemAssociation OwnerHistory Name Description Related Objects Related Objects	UtlMaterialProfileGet UtlMaterialProfileGet Has External References Has Properties Name Description Material Profiles Component Profile	UtlMaterialProfile UtlMaterialProfile Has External References Has Properties Name Description Material Profile Priority Category Category	UtlMaterial UtlMaterial Has External References Has Properties Name Description Category Material Has Representation Related Objects Related Objects	UtlMaterial Properties UtlMaterialProperties Name Description Properties Material	UtlMaterial Material Properties UtlMaterialMaterialProperties Name Description Representations Representations/Related	UtlPriority UtlPriority Name Description PartOf/Map Property/DependsOn Property/DependsOn PartOf/Related	UtlStyleItem UtlStyleItem Style/Assignment Style/Item Item Styles Name	UtlPresentationStyle UtlPresentationStyle Name
---	---	---	--	---	---	--	--	---	---

Elements requiring surrounding space for clearance provide a 'Clearance' representation. The reason for clearance space may be due to ventilation, maintenance, or other purpose. Examples of such elements include boilers and chillers. Such representation may be used for interference checks, where the 'Clearance' representation must not intersect with the 'Body' representation of other objects, though may intersect with the 'Clearance' representation of other objects.



Lighting Geometry

Elements emitting light provide a 'Lighting' representation. Examples of such elements include lamps and light fixtures. Such representation may be used for 3D rendering or lighting design.



4.7.6.4 Implementation

4.7.6.4.1 MVD schema listing

The schema encapsulating the data definitions for this model view is published in multiple representations.

An MVXML file defines the referenced entities and rules for this model view. This file may be used to validate instance data (in IFC-SPF or IFC-XML files), filter instance data to include entities and attributes within scope of this model view, or generate sub-schemas (including the EXP and XSD representations).

An EXP file represents the schema in EXPRESS format (ISO 10303-11) which adapts the referenced Industry Foundation Classes schema (ISO 16739) by including a subset of data definitions and a subset of attributes within each data definition. The EXP file may be used by software development tools for generating programming languages schemas (e.g. C++, C#, Java), database definitions (e.g. SQL DDL), and data transport schema definitions (e.g. XSD).

An XSD file represents the schema in XML Data Definition Language (XSD) which adapts the referenced subset of data definitions. The XSD file may be used by software development tools (e.g. Eclipse, Microsoft Visual Studio) to validate XML files and generate language-specific classes.

An IFC file represents the dynamic portions of the schema in the form of property sets within an SPF (ISO 10303-21) instance file.

The rationale for publishing multiple representations is to provide the richest level of integration for different implementations; while XSD is often used in defining web standards replacing document-based exchanges (e.g. invoices), it lacks data model information needed for type safety, data integrity, indexing, and optimization; all of which may be derived from the EXPRESS representation.

File	Format
SPARKie.exp	EXPRESS schema definition

File	Format
SPARKie.xsd	XML schema definition (XSD)
SPARKie.mvdxml	MVDXML schema transform
SPARKie.ifc	IFC dynamic schema definition

4.7.6.4.2 MVD format description

Implementations of this model view may publish instance data in various formats. Such format indicates the data encoding and does not necessarily imply that data may only be exchanged using physical files on computers; formats may be transmitted over the Internet as the “presentation layer” (OSI Layer 6) of any API. As the IFC data model supports both full and partial data models where all objects can be tagged to indicate merge directives (Create/Update/Delete using `IfcOwnerHistory.ChangeAction`), data may be transmitted in whole or in part, such as indicating only data changes.

As other OSI layers are already standardized, a full web API may be defined by referencing each layer as follows:

OSI Layer	OSI Layer Name	Protocol	Description
7	Application	WebDav	Defines valid operations such as GET, PUT, POST, DELETE, MKCOL, LOCK, UNLOCK
6	Presentation	IFC-SPF/IFC-XML	Defines data encoding
5	Session	HTTP/HTTPS	Defines establishment of sessions, compression, authentication, requests, responses, and errors
4	Transport	TCP	Defines message delivery
3	Network	IP	Defines network paths across multiple nodes
2	Data Link	MAC	Defines data frame communications between two nodes
1	Physical	(undefined)	Defines physical connectivity

Each supported format is listed by name, with Extension indicating the default file extension to use on applicable platforms (e.g. Windows), MIME type for indicating the HTTP header when transmitting over the Internet, and Reference standard indicating the presentation layer encoding format.

Format	Extension	MIME	Reference
IFC-SPF	.ifc	application/step	ISO 10303-21
IFC-XML	.ifcxml	application/xml	ISO 10303-28

IFC-SPF (ISO 10303-21) is a text format optimized to carry data with complex relationships, supporting human readability yet more compact representation (typically around 10% of size of equivalent XML).

IFC-HDF (ISO 10303-26) is a binary file format encapsulating data in a compact, indexable encoding optimized for quick retrieval and minimal memory usage.

NOTE As this file type is not yet widely implemented, it is not officially part of this model view, however implementations may prefer such format for internal use.

IFC-XML (ISO 10303-28) is a hierarchical markup format with wide support from software development tools and platforms, supporting greater human readability at the expense of larger representation.

NOTE As typical buildings contain millions of elements with graphs of relationships resulting in gigabytes of data, XML is not yet suitable for representing complete buildings from a pragmatic standpoint of data size, transmission cost, and loading time. However, using derived formats along with MVDXML to filter data sets may enable more efficient exchanges to take place.

IFC-ZIP (ISO 21320-1) is a compressed file format encapsulating one of the above formats to minimize data size.

NOTE As this model view is primarily intended for web-based exchange, zip compression may be selected by other means according to the client and server; therefore, the IFC-ZIP format is not officially part of this model view.

4.7.6.4.3 MVD dynamic schema analysis

Portions of data definitions are defined dynamically, to allow software applications to support extensible definitions while minimizing implementation overhead. Each property set is shown within a subsection as follows, with rows corresponding to properties. See *IfcPropertySet* for usage information.

Pset_ActorCommon

Property	Property Type	Data Type	Description
NumberOfActors	P_SINGLEVALUE	IfcCountMeasure	The number of actors that are to be dealt with together in the population.
Category	P_SINGLEVALUE	IfcLabel	Designation of the category into which the actors in the population belong.
SkillLevel	P_SINGLEVALUE	IfcLabel	Skill level exhibited by the actor and which indicates an extent of their capability to perform actions on the artefacts upon which they can act.

Pset_BuildingCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'). Used to store the non-classification driven internal project type.
BuildingID	P_SINGLEVALUE	IfcIdentifier	A unique identifier assigned to a building. A temporary identifier is initially assigned at the time of making a planning application. This temporary identifier is changed to a permanent identifier when the building is registered into a statutory buildings and properties database.
IsPermanentID	P_SINGLEVALUE	IfcBoolean	Indicates whether the identity assigned to a building is permanent (= TRUE) or temporary (=FALSE).
ConstructionMethod	P_SINGLEVALUE	IfcLabel	The type of construction action to the building, the project deals with, e.g. new construction, renovation, refurbishment, etc.
FireProtectionClass	P_SINGLEVALUE	IfcLabel	Main fire protection class for the building which is assigned from the fire protection classification table as given by the relevant national building code.

Property	Property Type	Data Type	Description
SprinklerProtection	P_SINGLEVALUE	IfcBoolean	Indication whether this object is sprinkler protected (TRUE) or not (FALSE).
SprinklerProtectionAutomatic	P_SINGLEVALUE	IfcBoolean	Indication whether this object has an automatic sprinkler protection (TRUE) or not (FALSE).
OccupancyType	P_SINGLEVALUE	IfcLabel	Occupancy type for this object. It is defined according to the presiding national building code.
GrossPlannedArea	P_SINGLEVALUE	IfcAreaMeasure	Total planned gross area for the building Used for programming the building.
NetPlannedArea	P_SINGLEVALUE	IfcAreaMeasure	Total planned net area for the building Used for programming the building.
NumberOfStoreys	P_SINGLEVALUE	IfcInteger	The number of storeys within a building. Captured for those cases where the IfcBuildingStorey entity is not used. Note that if IfcBuildingStorey is asserted and the number of storeys in a building can be determined from it, then this approach should be used in preference to setting a property for the number of storeys.
YearOfConstruction	P_SINGLEVALUE	IfcLabel	Year of construction of this building, including expected year of completion.
YearOfLastRefurbishment	P_SINGLEVALUE	IfcLabel	Year of last major refurbishment, or reconstruction, of the building (applies to reconstruction works).
IsLandmarked	P_SINGLEVALUE	IfcLogical	This building is listed as a historic building (TRUE), or not (FALSE), or unknown.

Pset_BuildingStoreyCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'). Used to store the non-classification driven internal project type.
EntranceLevel	P_SINGLEVALUE	IfcBoolean	Indication whether this building storey is an entrance level to the building (TRUE), or (FALSE) if otherwise.
AboveGround	P_SINGLEVALUE	IfcLogical	Indication whether this building storey is fully above ground (TRUE), or below ground (FALSE), or partially above and below ground (UNKNOWN) - as in sloped terrain.
SprinklerProtection	P_SINGLEVALUE	IfcBoolean	Indication whether this object is sprinkler protected (TRUE) or not (FALSE).
SprinklerProtectionAutomatic	P_SINGLEVALUE	IfcBoolean	Indication whether this object has an automatic sprinkler protection (TRUE) or not (FALSE). It should only be given, if the property "SprinklerProtection" is set to TRUE.
LoadBearingCapacity	P_SINGLEVALUE	IfcPlanarForceMeasure	Maximum load bearing capacity of the floor structure throughout the storey as designed.
GrossPlannedArea	P_SINGLEVALUE	IfcAreaMeasure	Total planned area for the building storey. Used for programming the building storey.
NetPlannedArea	P_SINGLEVALUE	IfcAreaMeasure	Total planned net area for the building storey. Used for programming the building storey.

Pset_SpaceCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'). Used to store the non-classification driven internal project type.

Property	Property Type	Data Type	Description
IsExternal	P_SINGLEVALUE	IfcBoolean	Indication whether the element is designed for use in the exterior (TRUE) or not (FALSE). If (TRUE) it is an external element and faces the outside of the building.
GrossPlannedArea	P_SINGLEVALUE	IfcAreaMeasure	Total planned gross area for the space. Used for programming the space.
NetPlannedArea	P_SINGLEVALUE	IfcAreaMeasure	Total planned net area for the space. Used for programming the space.
PubliclyAccessible	P_SINGLEVALUE	IfcBoolean	Indication whether this space (in case of e.g., a toilet) is designed to serve as a publicly accessible space, e.g., for a public toilet (TRUE) or not (FALSE).
HandicapAccessible	P_SINGLEVALUE	IfcBoolean	Indication whether this space (in case of e.g., a toilet) is designed to serve as an accessible space for handicapped people, e.g., for a public toilet (TRUE) or not (FALSE). This information is often used to declare the need for access for the disabled and for special design requirements of this space.

Pset_SpaceLightingRequirements

Property	Property Type	Data Type	Description
ArtificialLighting	P_SINGLEVALUE	IfcBoolean	Indication whether this space requires artificial lighting (as natural lighting would be not sufficient). (TRUE) indicates yes (FALSE) otherwise.
Illuminance	P_SINGLEVALUE	IfcIlluminanceMeasure	Required average illuminance value for this space.

Pset_SpaceOccupancyRequirements

Property	Property Type	Data Type	Description
OccupancyType	P_SINGLEVALUE	IfcLabel	Occupancy type for this object. It is defined according to the presiding national building code.
OccupancyNumber	P_SINGLEVALUE	IfcCountMeasure	Number of people required for the activity assigned to this space.
OccupancyNumberPeak	P_SINGLEVALUE	IfcCountMeasure	Maximal number of people required for the activity assigned to this space in peak time.
OccupancyTimePerDay	P_SINGLEVALUE	IfcTimeMeasure	The amount of time during the day that the activity is required within this space.
AreaPerOccupant	P_SINGLEVALUE	IfcAreaMeasure	Design occupancy loading for this type of usage assigned to this space.
MinimumHeadroom	P_SINGLEVALUE	IfcLengthMeasure	Headroom required for the activity assigned to this space.
IsOutlookDesirable	P_SINGLEVALUE	IfcBoolean	An indication of whether the outlook is desirable (set TRUE) or not (set FALSE)

SPARKie_SpaceElectricalRequirements

Property	Property Type	Data Type	Description
LightingPowerDensity	P_SINGLEVALUE	IfcEnergyMeasure	<p>Lighting Power Density (LPD) is a lighting power requirement defined in North America by the American National Standards Institute (ANSI), American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), and the Illuminating Engineering Society of North America (IESNA) Lighting subcommittee.</p> <p>Lighting Power Density technically represents the load of any lighting equipment in any defined area, commonly represented in watts per square foot of the lighting equipment. However, in the lighting industry it is often associated with the lighting power allowance (LPA) permitted by the building energy code in question.</p> <p>As no measurement unit is currently defined in IFC for</p>

			Energy/Area, units are expressed as Energy where the area is presumed to be the default project units for Area.
AppliancePowerDensity	P_SINGLEVALUE	IfcPowerMeasure	The load of appliances in any defined area, commonly represented in watts per square foot.
EquipmentPowerDensity	P_SINGLEVALUE	IfcPowerMeasure	The load of equipment in any defined area, commonly represented in watts per square foot.

Pset_DistributionPortTypeCable

Property	Property Type	Data Type	Description
ConnectionType	P_ENUMERATEDVALUE	IfcLabel/PEnum_DistributionPortElectricalType: ACPLUG, DCPLUG, CRIMPCOAXIAL, RJ, RADIO, DIN, DSUB, DVI, EIAJ, HDMI, RCA, TRS, XLR, OTHER, NOTKNOWN, UNSET	The physical port connection: ACPLUG: AC plug DCPLUG: DC plug CRIMP: bare wire
ConnectionSubtype	P_SINGLEVALUE	IfcLabel	The physical port connection subtype that further qualifies the ConnectionType. The following values are recommended: ACPLUG: A, B, C, D, E, F, EF, G, H, I, J, K, L, M; RADIO: IEEE802.11g, IEEE802.11n ; RJ: 4P4C, 6P2C, 8P8C; DIN: Mini3P, Mini4P, Mini5P, Mini6P, Mini7P, Mini8P, Mini9P DSub: DA15, DB25, DC37, DD50, DE9, DE15 EIAJ: RC5720 HDMI: A, B, C TRS: TS_Mini, TS_SubMini, TRS_Mini, TRS_SubMini
ConnectionGender	P_ENUMERATEDVALUE	IfcLabel/PEnum_DistributionPortGender: MALE, FEMALE, OTHER, NOTKNOWN, UNSET	The physical connection gender.
ConductorFunction	P_ENUMERATEDVALUE	IfcLabel/PEnum_ConductorFunctionEnum: PHASE_L1, PHASE_L2, PHASE_L3, NEUTRAL, PROTECTIVEEARTH, PROTECTIVEEARTHNEUTRAL, OTHER, NOTKNOWN, UNSET	For ports distributing power, indicates function of the conductors to which the load is connected.
CurrentContent3rdHarmonic	P_SINGLEVALUE	IfcPositiveRatioMeasure	The ratio between the third harmonic current and the phase current.
Current	P_BOUNDEDVALUE	IfcElectricCurrentMeasure	The actual current and operable range.
Voltage	P_BOUNDEDVALUE	IfcElectricVoltageMeasure	The actual voltage and operable range.
Power	P_BOUNDEDVALUE	IfcPowerMeasure	The actual power and operable range.
Protocols	P_LISTVALUE	IfcIdentifier	For data ports, identifies the protocols used as defined by the Open System Interconnection (OSI) Basic Reference Model (ISO 7498). Layers include: 1. Physical; 2. DataLink; 3. Network; 4. Transport; 5. Session; 6. Presentation; 7. Application. Example: 3: IP, 4: TCP, 5: HTTP

Pset_DistributionSystemCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specific instance of a distribution system, or sub-system (e.g. 'WWS/VS1', which indicates the system to be WWS, subsystems VSI/400). The reference values depend on the local code of practice.

Pset_DistributionSystemTypeElectrical

Property	Property Type	Data Type	Description
ElectricalSystem Type	P_ENUMERATEDVALUE	IfcLabel/PEnum_DistributionSystemElectricalType: TN, TN_C, TN_S, TN_C_S, TT, IT, OTHER, NOTKNOWN, UNSET	For certain purposes of electrical regulations, IEC 60364 defines types of system using type identifiers. Assignment of identifiers depends upon the relationship of the source, and of exposed conductive parts of the installation, to Ground (Earth). Identifiers that may be assigned through IEC 60364 are: •TN type system, a system having one or more points of the source of energy directly earthed, the exposed conductive parts of the installation being connected to that point by protective conductors, •TN C type system, a TN type system in which neutral and protective functions are combined in a single conductor throughout the system, •TN S type system, a TN type system having separate neutral and protective conductors throughout the system, •TN C S type system, a TN type system in which neutral and protective functions are combined in a single conductor in part of the system, •TT type system, a system having one point of the source of energy directly earthed, the exposed conductive parts of the installation being connected to earth electrodes electrically independent of the earth electrodes of the source, •IT type system, a system having no direct connection between live parts and Earth, the exposed conductive parts of the electrical installation being earthed.
ElectricalSystem Category	P_ENUMERATEDVALUE	IfcLabel/PEnum_DistributionSystemElectricalCategory: HIGHVOLTAGE, LOWVOLTAGE, EXTRALOWVOLTAGE, OTHER, NOTKNOWN, UNSET	Designates the voltage range of the circuit, according to IEC. HIGHVOLTAGE indicates >1000V AC or >1500V DV; LOWVOLTAGE indicates 50-1000V AC or 120-1500V DC; EXTRALOWVOLTAGE indicates <50V AC or <120V DC.
Diversity	P_SINGLEVALUE	IfcPositiveRatioMeasure	The ratio, expressed as a numerical value or as a percentage, of the simultaneous maximum demand of a group of electrical appliances or consumers within a specified period, to the sum of their individual maximum demands within the same period. The group of electrical appliances is in this case connected to this circuit. Definition from IEC 60050, IEC 691-10-04 NOTE1: It is often not desirable to size each conductor in a distribution system to support the total connected load at that point in the network. Diversity is applied on the basis of the anticipated loadings that are likely to result from all loads not being connected at the same time. NOTE2: Diversity is applied to final circuits only, not to sub-main circuits supplying other DBs.
NumberOfLiveConductors	P_SINGLEVALUE	IfcInteger	Number of live conductors within this circuit. Either this property or the ConductorFunction property (if only one) may be asserted.
MaximumAllowedVoltageDrop	P_SINGLEVALUE	IfcElectricVoltageMeasure	The maximum voltage drop across the circuit that must not be exceeded. There are two voltage drop limit settings that may be applied; one for sub-main circuits, and one in each Distribution Board or Consumer Unit for final circuits connected to that board. The settings should limit the overall voltage drop to the required level. Default settings of 1.5% for sub-main circuits and 2.5% for final circuits, giving an overall limit of 4% may be applied. NOTE: This value may also be specified as a constraint within an IFC model if required but is included within the property set at this stage pending implementation of the required capabilities within software applications.
NetImpedance	P_SINGLEVALUE	IfcElectricResistanceMeasure	The maximum earth loop impedance upstream of a circuit (typically stated as the variable Zs). This value is for 55°C (130°F) Celsius usage.

Pset_UtilityConsumptionPHistory

Property	Property Type	Data Type	Description
Heat	P_REFERENCEVALUE	IfcTimeSeries/IfcEnergyMeasure	The amount of heat energy consumed during the period specified in the time series.
Electricity	P_REFERENCEVALUE	IfcTimeSeries/IfcEnergyMeasure	The amount of electricity consumed during the period specified in the time series.
Water	P_REFERENCEVALUE	IfcTimeSeries/IfcVolumeMeasure	The amount of water consumed during the period specified in the time series.
Fuel	P_REFERENCEVALUE	IfcTimeSeries/IfcVolumeMeasure	The amount of fuel consumed during the period specified in the time series.
Steam	P_REFERENCEVALUE	IfcTimeSeries/IfcMassMeasure	The amount of steam consumed during the period specified in the time series.

Pset_CableCarrierFittingTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_CableCarrierSegmentTypeCableLadderSegment

Property	Property Type	Data Type	Description
NominalWidth	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal width of the segment.
NominalHeight	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal height of the segment.
LadderConfiguration	P_SINGLEVALUE	IfcText	Description of the configuration of the ladder structure used.

Pset_CableCarrierSegmentTypeCableTraySegment

Property	Property Type	Data Type	Description
NominalWidth	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal width of the segment.
NominalHeight	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal height of the segment.
HasCover	P_SINGLEVALUE	IfcBoolean	Indication of whether the cable tray has a cover (=TRUE) or not (=FALSE). By default, this value should be set to FALSE..

Pset_CableCarrierSegmentTypeCableTrunkingSegment

Property	Property Type	Data Type	Description
NominalWidth	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal width of the segment.
NominalHeight	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal height of the segment.
NumberOfCompartments	P_SINGLEVALUE	IfcInteger	The number of separate internal compartments within the trunking.

Pset_CableCarrierSegmentTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_CableCarrierSegmentTypeConduitSegment

Property	Property Type	Data Type	Description
NominalWidth	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal width of the segment.
NominalHeight	P_SINGLEVALUE	IfcPositiveLengthMeasure	The nominal height of the segment.
ConduitShapeType	P_ENUMERATEDVALUE	IfcLabel/PEnum_ConduitShapeType: CIRCULAR, OVAL, OTHER, NOTKNOWN, UNSET	The shape of the conduit segment.
IsRigid	P_SINGLEVALUE	IfcBoolean	Indication of whether the conduit is rigid (= TRUE) or flexible (= FALSE).

Pset_CableFittingTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_CableSegmentOccurrence

Property	Property Type	Data Type	Description
DesignAmbientTemperature	P_BOUNDEDVALUE	IfcThermodynamicTemperatureMeasure	The highest and lowest local ambient temperature likely to be encountered.
UserCorrectionFactor	P_SINGLEVALUE	IfcReal	An arbitrary correction factor that may be applied by the user.
NumberOfParallelCircuits	P_SINGLEVALUE	IfcInteger	Number of parallel circuits.
InstallationMethod	P_SINGLEVALUE	IfcLabel	Method of installation of cable/conductor. Installation methods are typically defined by reference in standards such as IEC 60364-5-52, table 52A-1 or BS7871 Appendix 4 Table 4A1 etc. Selection of the value to be used should be determined from such a standard according to local usage.
InstallationMethodFlagEnum	P_ENUMERATEDVALUE	IfcLabel/PEnum_InstallationMethodFlagEnum	Special installation conditions relating to particular types of installation based on IEC60364-5-52:2001 reference

Property	Property Type	Data Type	Description
		um: INDUCT, INSOIL, ONWALL, BELOWCEILING, OTHER, NOTKNOWN, UNSET	installation methods C and D.
DistanceBetweenParallelCircuits	P_SINGLEVALUE	IfcLengthMeasure	Distance measured between parallel circuits.
SoilConductivity	P_SINGLEVALUE	IfcThermalConductivityMeasure	Thermal conductivity of soil. Generally, within standards such as IEC 60364-5-52, table 52A-16, the resistivity of soil is required (measured in [SI] units of degK.m /W). This is the reciprocal of the conductivity value and needs to be calculated accordingly.
CarrierStackNumber	P_SINGLEVALUE	IfcInteger	Number of carrier segments (tray, ladder etc.) that are vertically stacked (vertical is measured as the z-axis of the local coordinate system of the carrier segment).
MountingMethod	P_ENUMERATEDVALUE	IfcLabel/PEnum_MountingMethodEnum: PERFORATEDTRAY, LADDER, OTHER, NOTKNOWN, UNSET	The method of mounting cable segment occurrences on a cable carrier occurrence from which the method required can be selected. This is for the purpose of carrying out 'worst case' cable sizing calculations and may be a conceptual requirement rather than a statement of the physical occurrences of cable and carrier segments.
IsHorizontalCable	P_SINGLEVALUE	IfcBoolean	Indication of whether the cable occurrences are mounted horizontally (= TRUE) or vertically (= FALSE).
IsMountedFlatCable	P_SINGLEVALUE	IfcBoolean	Indication of whether the cable occurrences are mounted flat (= TRUE) or in a trefoil pattern (= FALSE).
CurrentCarryingCapacity	P_SINGLEVALUE	IfcElectricCurrentMeasure	Maximum value of electric current which can be carried continuously by a conductor, a device or an apparatus, under specified conditions without its steady-state temperature exceeding a specified value. Based on IEC60826-11-13. NOTE: The temperature specified value is maximum Design Ambient Temperature.
MaximumCableLength	P_SINGLEVALUE	IfcLengthMeasure	Maximum cable length based on voltagedrop. NOTE: This value may also be specified as a constraint within an IFC model if required but is included within the property set at this stage pending implementation of the required capabilities within software applications.
PowerLoss	P_SINGLEVALUE	IfcElectricCurrentMeasure	Total loss of power across this cable.

Pset_CableSegmentTypeBusBarSegment

Property	Property Type	Data Type	Description
IsHorizontalBusbar	P_SINGLEVALUE	IfcBoolean	Indication of whether the busbar occurrences are routed horizontally (= TRUE) or vertically (= FALSE).

Pset_CableSegmentTypeCableSegment

Property	Property Type	Data Type	Description
Standard	P_SINGLEVALUE	IfcLabel	The designation of the standard applicable for the definition of the Cable/Bus used.
NumberOfCores	P_SINGLEVALUE	IfcInteger	The number of cores in Cable/Bus.
OverallDiameter	P_SINGLEVALUE	IfcPositiveLengthMeasure	The overall diameter of a Cable/Bus.
RatedVoltage	P_BOUNDEDVALUE	IfcElectricVolt	The range of allowed voltage that a device is

Property	Property Type	Data Type	Description
		ageMeasure	certified to handle. The upper bound of this value is the maximum.
RatedTemperature	P_BOUNDEDVALUE	IfcThermodynamicTemperatureMeasure	The range of allowed temperature that a device is certified to handle. The upper bound of this value is the maximum.
ScreenDiameter	P_SINGLEVALUE	IfcPositiveLengthMeasure	The diameter of the screen around a cable or bus segment (if present).
HasProtectiveEarth	P_SINGLEVALUE	IfcBoolean	One core has protective earth marked insulation, Yellow/Green.
MaximumOperatingTemperature	P_SINGLEVALUE	IfcThermodynamicTemperatureMeasure	The maximum temperature at which a cable or bus is certified to operate.
MaximumShortCircuitTemperature	P_SINGLEVALUE	IfcThermodynamicTemperatureMeasure	The maximum short circuit temperature at which a cable or bus is certified to operate.
SpecialConstruction	P_SINGLEVALUE	IfcLabel	Special construction capabilities like self-supporting, flat deidable cable or bus flat non deidable cable or bus supporting elements inside (steel, textile, concentric conductor). Note that materials used should be agreed between exchange participants before use.
Weight	P_SINGLEVALUE	IfcMassMeasure	Weight of cable kg/km.
SelfExtinguishing60332_1	P_SINGLEVALUE	IfcBoolean	Self Extinguishing cable/core according to IEC 60332.1.
SelfExtinguishing60332_3	P_SINGLEVALUE	IfcBoolean	Self Extinguishing cable/core according to IEC 60332.3.
HalogenProof	P_SINGLEVALUE	IfcBoolean	Produces small amount of smoke and irritating Deaerator/Gas.
FunctionReliable	P_SINGLEVALUE	IfcBoolean	Cable/bus maintain given properties/functions over a given (tested) time and conditions. According to IEC standard.

Pset_CableSegmentTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_CableSegmentTypeConductorSegment

Property	Property Type	Data Type	Description
CrossSectionalArea	P_SINGLEVALUE	IfcAreaMeasure	Cross section area of the phase(s) lead(s).
Function	P_ENUMERATEDVALUE	IfcLabel/PEnum_FunctionEnum: LINE, NEUTRAL, PROTECTIVE	Type of function for which the conductor is intended.

Property	Property Type	Data Type	Description
		EARTH,PROTECTIVEEARTHNEUTRAL,OTHER,NOTKNOWN,UNSET	
Material	P_ENUMERATEDVALUE	IfcLabel/PEnum_MaterialEnum: ALUMINIUM,COPPER,OTHER,NOTKNOWN,UNSET	Type of material from which the conductor is constructed.
Construction	P_ENUMERATEDVALUE	IfcLabel/PEnum_ConstructionEnum: SOLIDCONDUCTOR,STRANDEDCONDUCTOR,FLEXIBLESTRANDEDCONDUCTOR,OTHER,NOTKNOWN,UNSET	Purpose of informing on how the conductor is constructed (intertwined or solid). I.e. Solid (IEV 461-01-06), stranded (IEV 461-01-07), solid-/finestranded (IEV 461-01-11) (not flexible/flexible).
Shape	P_ENUMERATEDVALUE	IfcLabel/PEnum_ShapeEnum: HELICALCONDUCTOR,CIRCULARCONDUCTOR,SECTORCONDUCTOR,RECTANGULARCONDUCTOR,OTHER,NOTKNOWN,UNSET	Indication of the shape of the conductor.

Pset_CableSegmentTypeCoreSegment

Property	Property Type	Data Type	Description
OverallDiameter	P_SINGLEVALUE	IfcPositiveLengthMeasure	The overall diameter of a core (maximum space used).
RatedVoltage	P_BOUNDEDVALUE	IfcElectricVoltageMeasure	The range of allowed voltage that a device is certified to handle. The upper bound of this value is the maximum.
RatedTemperature	P_BOUNDEDVALUE	IfcThermodynamicTemperatureMeasure	The range of allowed temperature that a device is certified to handle. The upper bound of this value is the maximum.
ScreenDiameter	P_SINGLEVALUE	IfcPositiveLengthMeasure	The diameter of the screen around a core segment (if present).
CoreIdentifier	P_SINGLEVALUE	IfcIdentifier	The core identification used. Identifiers may be used such as by color (Black, Brown, Grey) or by number (1, 2, 3) or by IEC phase reference (L1, L2, L3) etc.
SheathColors	P_ENUMERATEDVALUE	IfcLabel/PEnum_CoreColorsEnum: BLACK,BLUE,BROWN,GOLD,GREEN,GREY,ORANGE,PINK,RED,SILVER,TURQUOISE,VIOLET,WHITE,YELLOW,GREENANDYELLOW,OTHER,NOTKNOWN,UNSET	Colour of the core (derived from IEC 60757). Note that the combined color 'GreenAndYellow' shall be used only as Protective Earth (PE) conductors according to the requirements of IEC 60446.
Weight	P_SINGLEVALUE	IfcMassMeasure	Weight of core kg/km.
SelfExtinguishing60332_1	P_SINGLEVALUE	IfcBoolean	Self Extinguishing cable/core according to IEC 60332.1.
SelfExtinguishing60332_3	P_SINGLEVALUE	IfcBoolean	Self Extinguishing cable/core according to IEC 60332.3.
HalogenProof	P_SINGLEVALUE	IfcBoolean	Produces small amount of smoke and irritating deaerator/gas.
FunctionReliable	P_SINGLEVALUE	IfcBoolean	Core maintain given properties/functions over a given (tested) time and conditions. According to (IEC) standard.
Standard	P_SINGLEVALUE	IfcLabel	The designation of the standard applicable for the definition of the core used.

Pset_ElectricalDeviceCommon

Property	Property Type	Data Type	Description
RatedCurrent	P_BOUNDEDVALUE	IfcElectricCurrentMeasure	The current that a device is designed to handle.
RatedVoltage	P_BOUNDEDVALUE	IfcElectricVoltageMeasure	The voltage that a device is designed to handle.
NominalFrequencyRange	P_BOUNDEDVALUE	IfcFrequencyMeasure	The upper and lower limits of frequency for which the operation of the device is certified.
PowerFactor	P_SINGLEVALUE	IfcNormalisedRatioMeasure	The ratio between the rated electrical power and the product of the rated current and rated voltage
ConductorFunction	P_ENUMERATEDVALUE	IfcLabel/PEnum_ConductorFunctionEnum: L1,L2,L3	Function of a line conductor to which a device is intended to be connected where L1, L2 and L3 represent the phase lines according to IEC 60446 notation (sometimes phase lines may be referenced by color [Red, Blue, Yellow] or by number [1, 2, 3] etc). Protective Earth is sometimes also known as CPC or common protective conductor. Note that for an electrical device, a set of line conductor functions may be applied.
NumberOfPoles	P_SINGLEVALUE	IfcInteger	The number of live lines that is intended to be handled by the device.
HasProtectiveEarth	P_SINGLEVALUE	IfcBoolean	Indicates whether the electrical device has a protective earth connection (=TRUE) or not (= FALSE).
IP_Code	P_SINGLEVALUE	IfcLabel	IEC 60529 (1989) Classification of degrees of protection provided by enclosures (IP Code).
InsulationStandardClass	P_ENUMERATEDVALUE	IfcLabel/PEnum_InsulationStandardClass: CLASS0APPLIANCE, CLASS1APPLIANCE, CLASS2APPLIANCE, CLASS3APPLIANCE, CLASS4APPLIANCE, CLASS5APPLIANCE, CLASS6APPLIANCE, CLASS7APPLIANCE, CLASS8APPLIANCE, CLASS9APPLIANCE, NOTKNOWN, UNSET	Insulation standard classes provides basic protection information against electric shock. Defines levels of insulation required in terms of constructional requirements (creepage and clearance distances) and electrical requirements (compliance with electric strength tests). Basic insulation is considered to be shorted under single fault conditions. The actual values required depend on the working voltage to which the insulation is subjected, as well as other factors. Also indicates whether the electrical device has a protective earth connection.

Pset_ElectricAppliancePHistory

Property	Property Type	Data Type	Description
PowerState	P_REFERENCEVALUE	IfcTimeSeries/IfcBoolean	Indicates the power state of the device where True is on and False is off.

Pset_ElectricApplianceTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_ElectricApplianceTypeDishwasher

Property	Property Type	Data Type	Description
DishwasherType	P_ENUMERATEDVALUE	IfcLabel/PEnum_ElectricApplianceDishwasherType: POTWASHER, TRAYWASHER, DISHWASHER, BOTTLEWASHER, CUTLERYWASHER, OTHER, UNKNOWN, UNSET	Type of dishwasher.

Pset_ElectricApplianceTypeElectricCooker

Property	Property Type	Data Type	Description
ElectricCookerType	P_ENUMERATEDVALUE	IfcLabel/PEnum_ElectricApplianceElectricCookerType: STEAMCOOKER, DEEPLY, STOVE, OVEN, TILTINGFRYINGPAN, COOKINGKETTLE, OTHER, UNKNOWN, UNSET	Type of electric cooker.

Pset_ElectricDistributionBoardOccurrence

Property	Property Type	Data Type	Description
IsMain	P_SINGLEVALUE	IfcBoolean	Identifies if the current instance is a main distribution point or topmost level in an electrical distribution hierarchy (= TRUE) or a sub-main distribution point (= FALSE).
IsSkilledOperator	P_SINGLEVALUE	IfcBoolean	Identifies if the current instance requires a skilled person or instructed person to perform operations on the distribution board (= TRUE) or whether operations may be performed by a person without appropriate skills or instruction (= FALSE).

Pset_ElectricDistributionBoardTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_ElectricFlowStorageDeviceTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).
NominalSupplyVoltage	P_SINGLEVALUE	IfcElectricVoltageMeasure	The nominal voltage of the supply.
NominalSupplyVoltageOffset	P_BOUNDEDVALUE	IfcElectricVoltageMeasure	The maximum and minimum allowed voltage of the supply e.g. boundaries of 380V/440V may be applied for a nominal voltage of 400V.

Property	Property Type	Data Type	Description
NominalFrequency	P_SINGLE VALUE	IfcFrequencyMeasure	The nominal frequency of the supply.
ConnectedConductor Function	P_ENUMERATED VALUE	IfcLabel/PEnum_ConductorFunctionEnum: PHASE_L1, PHASE_L2, PHASE_L3, NEUTRAL, PROTECTIVEEARTH, PROTECTIVEEARTHNEUTRAL, OTHER, NOTKNOWN, UNSET	Function of the conductors to which the load is connected.
ShortCircuit3PoleMaximumState	P_SINGLE VALUE	IfcElectricCurrentMeasure	Maximum 3 pole short circuit current provided at the point of supply.
ShortCircuit3PolePowerFactorMaximumState	P_SINGLE VALUE	IfcReal	Power factor of the maximum 3 pole short circuit current provided at the point of supply.
ShortCircuit2PoleMinimumState	P_SINGLE VALUE	IfcElectricCurrentMeasure	Minimum 2 pole short circuit current provided at the point of supply.
ShortCircuit2PolePowerFactorMinimumState	P_SINGLE VALUE	IfcReal	Power factor of the minimum 2 pole short circuit current provided at the point of supply.
ShortCircuit1PoleMaximumState	P_SINGLE VALUE	IfcElectricCurrentMeasure	Maximum 1 pole short circuit current provided at the point of supply i.e. the fault between 1 phase and N.
ShortCircuit1PolePowerFactorMaximumState	P_SINGLE VALUE	IfcReal	Power factor of the maximum 1 pole short circuit current provided at the point of supply i.e. the fault between 1 phase and N.
ShortCircuit1PoleMinimumState	P_SINGLE VALUE	IfcElectricCurrentMeasure	Minimum 1 pole short circuit current provided at the point of supply i.e. the fault between 1 phase and N.
ShortCircuit1PolePowerFactorMinimumState	P_SINGLE VALUE	IfcReal	Power factor of the minimum 1 pole short circuit current provided at the point of supply i.e. the fault between 1 phase and N.
EarthFault1PoleMaximumState	P_SINGLE VALUE	IfcElectricCurrentMeasure	Maximum 1 pole earth fault current provided at the point of supply i.e. the fault between 1 phase and PE/PEN.
EarthFault1PolePowerFactorMaximumState	P_SINGLE VALUE	IfcReal	Power factor of the maximum 1 pole earth fault current provided at the point of supply i.e. the fault between 1 phase and PE/PEN.
EarthFault1PoleMinimumState	P_SINGLE VALUE	IfcElectricCurrentMeasure	Minimum 1 pole earth fault current provided at the point of supply i.e. the fault between 1 phase and PE/PEN.
EarthFault1PolePowerFactorMinimumState	P_SINGLE VALUE	IfcReal	Power factor of the minimum 1 pole earth fault current provided at the point of supply i.e. the fault between 1 phase and PE/PEN.

Pset_ElectricGeneratorTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLE VALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATED VALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

ElectricGeneratorEfficiency	P_SINGLEVALUE	IfcPositiveRatioMeasure	The ratio of output capacity to intake capacity.
StartCurrentFactor	P_SINGLEVALUE	IfcReal	IEC. Start current factor defines how large the peak starting current will become on the engine. StartCurrentFactor is multiplied to NominalCurrent and we get the start current.
MaximumPowerOutput	P_SINGLEVALUE	IfcPowerMeasure	The maximum output power rating of the engine.

Pset_JunctionBoxTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).
NumberOfGangs	P_SINGLEVALUE	IfcInteger	Number of slots available for switches/outlets (most commonly 1, 2, 3, or 4).
ClearDepth	P_SINGLEVALUE	IfcPositiveLengthMeasure	Clear unobstructed depth available for cable inclusion within the junction box.
ShapeType	P_ENUMERATEDVALUE	IfcLabel/PEnum_JunctionBoxShapeType: RECTANGULAR, ROUND, SLOT, OTHER, NOTKNOWN, UNSET	Shape of the junction box.
PlacingType	P_ENUMERATEDVALUE	IfcLabel/PEnum_JunctionBoxPlacingType: CEILING, FLOOR, WALL, OTHER, NOTKNOWN, UNSET	Location at which the type of junction box can be located.
MountingType	P_ENUMERATEDVALUE	IfcLabel/PEnum_JunctionBoxMountingType: FACENAIL, SIDENAIL, CUT_IN, OTHER, NOTKNOWN, UNSET	Method of mounting to be adopted for the type of junction box.
IsExternal	P_SINGLEVALUE	IfcBoolean	Indication of whether the junction box type is allowed for exposure to outdoor elements (set TRUE where external exposure is allowed).
IP_Code	P_SINGLEVALUE	IfcLabel	IEC 60529 (1989) Classification of degrees of protection provided by enclosures (IP Code).

Pset_LightFixtureTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).
NumberOfSources	P_SINGLE	IfcInteger	Number of sources .

rces	VALUE		
TotalWattage	P_SINGLE VALUE	IfcPowerMeasure	Wattage on whole lightfitting device with all sources intact.
LightFixtureMountingType	P_ENUMERATED VALUE	IfcLabel/PEnum_LightFixtureMountingType: CABLESPANNE D,FREESTANDING,POLE_SID E,POLE_TOP,RECESSED,SURFACE,SUSPENDED,TRACKMOUNTED,OTHER,NOTKNOWN,UNSET	A list of the available types of mounting for light fixtures from which that required may be selected.
LightFixturePlacingType	P_ENUMERATED VALUE	IfcLabel/PEnum_LightFixturePlacingType: CEILING,FLOOR,FURNITURE,POLE,WALL,OTHER,NOTKNOWN,UNSET	A list of the available types of placing specification for light fixtures from which that required may be selected.
MaintenanceFactor	P_SINGLE VALUE	IfcReal	The arithmetical allowance made for depreciation of lamps and reflective equipment from their initial values due to dirt, fumes, or age.
MaximumPlenumSensibleLoad	P_SINGLE VALUE	IfcPowerMeasure	Maximum or Peak sensible thermal load contributed to the conditioned space by the light fixture.
MaximumSpaceSensibleLoad	P_SINGLE VALUE	IfcPowerMeasure	Maximum or Peak sensible thermal load contributed to return air plenum by the light fixture.
SensibleLoadToRadiant	P_SINGLE VALUE	IfcPositiveRatioMeasure	Percent of sensible thermal load to radiant heat.

Pset_LightFixtureTypeSecurityLighting

Property	Property Type	Data Type	Description
SecurityLightingType	P_ENUMERATED VALUE	IfcLabel/PEnum_LightFixtureSecurityLightingType: SAFETYLIGHT,WARNINGLIGHT,EMERGENCYEXITLIGHT,BLUEILLUMINATION,OTHER,NOTKNOWN,UNSET	The type of security lighting.
FixtureHeight	P_SINGLE VALUE	IfcPositiveLengthMeasure	The height of the fixture, such as the text height of an exit sign.
SelfTestFunction	P_ENUMERATED VALUE	IfcLabel/PEnum_SelfTestType: CENTRAL,LOCAL,NONE,OTHER,NOTKNOWN,UNSET	The type of self test function.
BackupSupplySystem	P_ENUMERATED VALUE	IfcLabel/PEnum_BackupSupplySystemType: LOCALBATTERY,CENTRALBATTERY,OTHER,NOTKNOWN,UNSET	The type of backup supply system.
PictogramEscapeDirection	P_ENUMERATED VALUE	IfcLabel/PEnum_PictogramEscapeDirectionType: RIGHTARROW,LEFTARROW,DOWNARROW,UPARROW,OTHER,NOTKNOWN,UNSET	The direction of escape pictogram.
Addressability	P_ENUMERATED VALUE	IfcLabel/PEnum_AddressabilityType: IMPLEMENTED,UPGRADEABLETO,NOTIMPLEMENTED,OTHER,NOTKNOWN,UNSET	The type of addressability.

Pset_OutletTypeCommon

Property	Property Type	Data Type	Description
----------	---------------	-----------	-------------

Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Statu s: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exists only temporary (like a temporary support structure).
IsPluggableOutlet	P_SINGLEVALUE	IfcLogical	Indication of whether the outlet accepts a loose plug connection (= TRUE) or whether it is directly connected (= FALSE) or whether the form of connection has not yet been determined (= UNKNOWN).

Pset_ProtectiveDeviceBreakerUnitI2TCurve

Property	Property Type	Data Type	Description
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230, U400, U440, U525, U690, U1000, OTHER, NOTKNOWN, UNSET	The voltage levels of the protective device for which the data of the instance is valid. More than one value may be selected in the enumeration.
NominalCurrent	P_SINGLEVALUE	IfcElectricCurrentMeasure	A set of nominal currents in [A] for which the data of this instance is valid. At least one value shall be provided. Any value in the set shall not exceed the value of the UltimateRatedCurrent associated with the same breaker unit.
BreakerUnitCurve	P_TABLEVALUE	IfcElectricCurrentMeasure/IfcReal	A curve that establishes the let through energy of a breaker unit when a particular prospective current is applied. Note that the breaker unit curve is defined within a Cartesian coordinate system and this fact must be asserted within the property set: (1) Defining value: ProspectiveCurrent: A list of minimum 2 and maximum 16 numbers providing the currents in [A] for points in the current/I2t log/log coordinate space. The curve is drawn as a straight line between two consecutive points. (2) Defined value: LetThroughEnergy: A list of minimum 2 and maximum 16 numbers providing the let-through energy, I2t, in [A2s] for points in the current/I2t log/log coordinate space. The curve is drawn as a straight line between two consecutive points.

Pset_ProtectiveDeviceBreakerUnitI2TFuseCurve

Property	Property Type	Data Type	Description
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230, U400, U440, U525, U690, U1000, OTHER, NOTKNOWN, UNSET	The voltage levels of the fuse for which the data of the instance is valid. More than one value may be selected in the enumeration.
BreakerUnitFuseMeltingCurve	P_TABLEVALUE	IfcElectricCurrentMeasure/IfcReal	A curve that establishes the energy required to melt the fuse of a breaker unit when a particular prospective melting current is applied. Note that the breaker unit fuse melting curve is defined within a Cartesian coordinate system and this fact must be: (1) Defining value: ProspectiveCurrentMelting : A list of minimum 2 and maximum 8 numbers providing the currents in [A] for points in the current/melting_energy log/log coordinate space. The curve is drawn as a straight line between two consecutive points. (2) Defined value: MeltingEnergy: A list of minimum 2 and maximum 8 numbers providing the energy whereby the fuse is starting to melt, I2t, in [A2s] for points in the current/melting_energy log/log coordinate space. The curve is drawn as a straight line between two consecutive points.

Property	Property Type	Data Type	Description
BreakerUnitFuseBreakingCurve	P_TABLEVALUE	IfcElectricCurrentMeasure/IfcReal	A curve that establishes the let through breaking energy of a breaker unit when a particular prospective breaking current is applied. Note that the breaker unit fuse breaking curve is defined within a Cartesian coordinate system and this fact must be: (1) Defining value: ProspectiveCurrentBreaking: A list of minimum 2 and maximum 8 numbers providing the currents in [A] for points in the current/breaking energy log/log coordinate space. The curve is drawn as a straight line between two consecutive points. (2) Defined value: LetThroughBreakingEnergy: A list of minimum 2 and maximum 8 numbers providing the breaking energy whereby the fuse has provided a break, I2t, in [A2s] for points in the current/breaking_energy log/log coordinate space. The curve is drawn as a straight line between two consecutive.

Pset_ProtectiveDeviceBreakerUnitIPICurve

Property	Property Type	Data Type	Description
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230, U400, U440, U525, U690, U1000, OTHER, NOTKNOWN, UNSET	The voltage level of the protective device for which the data of the instance is valid. More than one value may be selected in the enumeration.
NominalCurrent	P_SINGLEVALUE	IfcElectricCurrentMeasure	A set of nominal currents in [A] for which the data of this instance is valid. At least one value shall be provided. Any value in the set shall not exceed the value of the UltimateRatedCurrent associated with the same breaker unit.
BreakerUnitIPICurve	P_TABLEVALUE	IfcElectricCurrentMeasure/IfcElectricCurrentMeasure	A curve that establishes the let through peak current of a breaker unit when a particular prospective current is applied. Note that the breaker unit IPI curve is defined within a Cartesian coordinate system and this fact must be asserted within the property set: (1) Defining value: A list of minimum 2 and maximum 16 numbers providing the currents in [A] for points in the I/Ī log/log coordinate space. The curve is drawn as a straight line between two consecutive points. (2) Defined value: A list of minimum 2 and maximum 16 numbers providing the let-through peak currents, Ī, in [A] for points in the I/Ī log/log coordinate space. The curve is drawn as a straight line between two consecutive points.

Pset_ProtectiveDeviceBreakerUnitTypeMCB

Property	Property Type	Data Type	Description
PowerLoss	P_SINGLEVALUE	IfcPowerMeasure	The power loss in [W] per pole of the MCB when the nominal current is flowing through the MCB.
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230, U400, U440, U525, U690, U1000, OTHER, NOTKNOWN, UNSET	The voltage levels for which the data of the instance is valid. More than one value may be selected in the enumeration.
NominalCurrents	P_LISTVALUE	IfcElectricCurrentMeasure	A set of nominal currents in [A] for which the data of this instance is valid. At least one value shall be provided. Any value in the set shall not exceed the value of the UltimateRatedCurrent associated with the same breaker unit.
ICU60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The ultimate breaking capacity in [A] for an MCB tested in accordance with the IEC 60947 series.
ICS60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The service breaking capacity in [A] for an MCB tested in

Property	Property Type	Data Type	Description
			accordance with the IEC 60947 series.
ICN60898	P_SINGLEVALUE	IfcElectricCurrentMeasure	The nominal breaking capacity in [A] for an MCB tested in accordance with the IEC 60898 series.
ICS60898	P_SINGLEVALUE	IfcElectricCurrentMeasure	The service breaking capacity in [A] for an MCB tested in accordance with the IEC 60898 series.

Pset_ProtectiveDeviceBreakerUnitTypeMotorProtection

Property	Property Type	Data Type	Description
PerformanceClasses	P_LISTVALUE	IfcLabel	A set of designations of performance classes for the breaker unit for which the data of this instance is valid. A breaker unit being a motor protection device may be constructed for different levels of breaking capacities. A maximum of 7 different performance classes may be provided. Examples of performance classes that may be specified include B, C, N, S, H, L, V.
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230,U400,U440,U525,U690,U1000,OTHER,NOTKNOWN,UNSET	The voltage levels for which the data of the instance is valid. More than one value may be selected in the enumeration.
ICU60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The ultimate breaking capacity in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series.
ICS60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The service breaking capacity in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series.
ICW60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The thermal withstand current in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series. The value shall be related to 1 s.
ICM60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The making capacity in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series.

Pset_ProtectiveDeviceTrippingCurve

Property	Property Type	Data Type	Description
TrippingCurveType	P_ENUMERATEDVALUE	IfcLabel/PEnum_TrippingCurveType: UPPER,LOWER,OTHER,NOTKNOWN,UNSET	The type of tripping curve that is represented by the property set.
TrippingCurve	P_TABLEVALUE	IfcElectricCurrentMeasure/IfcTimeMeasure	A curve that establishes the release time of a tripping unit when a particular prospective current is applied. Note that the tripping curve is defined within a Cartesian coordinate system and this fact must be asserted within the property set: (1) Defining value is the Prospective Current which is a list of minimum 2 and maximum 16 numbers providing the currents in [x In] for points in the current/time log/log coordinate space. The curve is drawn as a straight line between two consecutive points. (2) Defined value is a list of minimum 2 and maximum 16 numbers providing the release_time in [s] for points in the current/time log/log coordinate space. The curve is drawn as a straight line between two consecutive points. Note that a defined interpolation.

Pset_ProtectiveDeviceTypeCircuitBreaker

Property	Property Type	Data Type	Description
PerformanceClasses	P_LISTVALUE	IfcLabel	A set of designations of performance classes for the breaker unit for which the data of this instance is valid. A breaker unit being a circuit breaker may be constructed for different levels of breaking capacities. A maximum of 7 different performance classes may be provided. Examples of performance classes that may be specified include B, C, N, S, H, L, V.
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230,U400,U440,U525,U690,U1000,OTHER,NOTKNOWN,UNSET	The voltage levels for which the data of the instance is valid. More than one value may be selected in the enumeration.
ICU60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The ultimate breaking capacity in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series.
ICS60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The service breaking capacity in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series.
ICW60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The thermal withstand current in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series. The value shall be related to 1 s.
ICM60947	P_SINGLEVALUE	IfcElectricCurrentMeasure	The making capacity in [A] for a circuit breaker or motor protection device tested in accordance with the IEC 60947 series.

Pset_ProtectiveDeviceTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_ProtectiveDeviceTypeEarthLeakageCircuitBreaker

Property	Property Type	Data Type	Description
EarthFailureDeviceType	P_ENUMERATEDVALUE	IfcLabel/PEnum_EarthFailureDeviceType: STANDARD, TIMEDELAYED, OTHER, NOTKNOWN, UNSET	A list of the available types of circuit breaker from which that required may be selected where: Standard: Device that operates without a time delay. TimeDelayed: Device that operates after a time delay.
Sensitivity	P_SINGLEVALUE	IfcElectricCurrentMeasure	The rated rms value of the vector sum of the instantaneous currents flowing in the main circuits of the device which causes the device to operate under specified conditions. (IEC 61008-1).

Pset_ProtectiveDeviceTypeFuseDisconnecter

Property	Property Type	Data Type	Description
FuseDisco	P_ENUME	IfcLabel/PEnum_FuseDisconnecterT	A list of the available types of fuse disconnecter from which

Property	Property Type	Data Type	Description
nnectorType	RATEDVALUE	ype: ENGINEPROTECTIONDEVICE, FUSE, SWITCH, HRC, OVERLOADPROTECTIONDEVICE, SWITCHDISCONNECTOR, FUSE, OTHER, NOTKNOWN, UNSET	that required may be selected where: EngineProtectionDevice: A fuse whose characteristic is specifically designed for the protection of a motor or generator. FuseSwitchDisconnect: A switch disconnect in which a fuse link or a fuse carrier with fuse link forms the moving contact, HRC: A standard fuse (High Rupturing Capacity) OverloadProtectionDevice: A device that disconnects the supply when the operating conditions in an electrically undamaged circuit causes an overcurrent, SemiconductorFuse: A fuse whose characteristic is specifically designed for the protection of semiconductor devices. SwitchDisconnectFuse: A switch disconnect in which one or more poles have a fuse in series in a composite unit.
VoltageLevel	P_ENUMERATEDVALUE	IfcLabel/PEnum_VoltageLevels: U230, U400, U440, U525, U690, U1000, OTHER, NOTKNOWN, UNSET	The voltage levels for which the data of the instance is valid. More than one value may be selected in the enumeration.
IC60269	P_SINGLEVALUE	IfcElectricCurrentMeasure	The breaking capacity in [A] for fuses in accordance with the IEC 60269 series.
PowerLoss	P_SINGLEVALUE	IfcPowerMeasure	The power loss in [W] of the fuse when the nominal current is flowing through the fuse.

Pset_ProtectiveDeviceTypeResidualCurrentCircuitBreaker

Property	Property Type	Data Type	Description
Sensitivity	P_SINGLEVALUE	IfcElectricCurrentMeasure	Current leakage to an unwanted leading path during normal operation (IEC 151-14-49).

Pset_ProtectiveDeviceTypeResidualCurrentSwitch

Property	Property Type	Data Type	Description
Sensitivity	P_SINGLEVALUE	IfcElectricCurrentMeasure	Current leakage to an unwanted leading path during normal operation (IEC 151-14-49).

Pset_ProtectiveDeviceTypeVaristor

Property	Property Type	Data Type	Description
VaristorType	P_ENUMERATEDVALUE	IfcLabel/PEnum_VaristorType: METALOXIDE, ZINCOXIDE, OTHER, NOTKNOWN, UNSET	A list of the available types of varistor from which that required may be selected.

Pset_SolarDeviceTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1'), provided, if there is no classification reference to a recognized classification system used.
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).

Pset_FlowMeterOccurrence

Property	Property Type	Data Type	Description
Purpose	P_ENUMERATEDVALUE	IfcLabel/PEnum_FlowMeterPurpose: MASTER, SUBMASTER, SUBMETER, OTHER, NOTKNOWN, UNSET	Enumeration defining the purpose of the flow meter occurrence.

Pset_FlowMeterTypeCommon

Property	Property Type	Data Type	Description
Reference	P_SINGLEVALUE	IfcIdentifier	Reference ID for this specified type in this project (e.g. type 'A-1').
Status	P_ENUMERATEDVALUE	IfcLabel/PEnum_Status: NEW, EXISTING, DEMOLISH, TEMPORARY, OTHER, NOTKNOWN, UNSET	Status of the element, predominately used in renovation or retrofitting projects. The status can be assigned to as "New" - element designed as new addition, "Existing" - element exists and remains, "Demolish" - element existed but is to be demolished, "Temporary" - element will exist only temporary (like a temporary support structure).
ReadOutType	P_ENUMERATEDVALUE	IfcLabel/PEnum_MeterReadOutType: DIAL, DIGITAL, OTHER, NOTKNOWN, UNSET	Indication of the form that readout from the meter takes. In the case of a dial read out, this may comprise multiple dials that give a cumulative reading and/or a mechanical odometer.
RemoteReading	P_SINGLEVALUE	IfcBoolean	Indicates whether the meter has a connection for remote reading through connection of a communication device (set TRUE) or not (set FALSE).

Pset_FlowMeterTypeEnergyMeter

Property	Property Type	Data Type	Description
NominalCurrent	P_SINGLEVALUE	IfcElectricCurrentMeasure	The nominal current that is designed to be measured.
MaximumCurrent	P_SINGLEVALUE	IfcElectricCurrentMeasure	The maximum allowed current that a device is certified to handle.
MultipleTarriff	P_SINGLEVALUE	IfcBoolean	Indicates whether meter has built-in support for multiple tarriffs (variable energy cost rates).

4.7.6.4.4 Non-applicable entity exclusion analysis

The referenced IFC schema is shown in the following table, with each row corresponding to a schema namespace, with data definitions listed within, where bold items indicate definitions within scope of this Model View Definition.

Namespace	Definitions	Usage
IfcKernel	IfcActor ; IfcComplexPropertyTemplate; IfcContext ; IfcControl ; IfcGroup ; IfcObject ; IfcObjectDefinition ; IfcPreDefinedPropertySet; IfcProcess ; IfcProduct ; IfcProject ; IfcProjectLibrary; IfcPropertyDefinition ; IfcPropertySet ; IfcPropertySetDefinition ; IfcPropertySetTemplate; IfcPropertyTemplate; IfcPropertyTemplateDefinition; IfcProxy; IfcQuantitySet; IfcRelAggregates ; IfcRelAssigns ; IfcRelAssignsToActor ; IfcRelAssignsToControl ; IfcRelAssignsToGroup ; IfcRelAssignsToGroupByFactor; IfcRelAssignsToProcess ; IfcRelAssignsToProduct ; IfcRelAssignsToResource ; IfcRelAssociates ; IfcRelAssociatesClassification ; IfcRelAssociatesDocument; IfcRelAssociatesLibrary; IfcRelationship ; IfcRelConnects ; IfcRelDeclares ; IfcRelDecomposes ; IfcRelDefines ; IfcRelDefinesByObject ; IfcRelDefinesByProperties ; IfcRelDefinesByTemplate; IfcRelDefinesByType ; IfcRelNests ; IfcResource ; IfcRoot ; IfcSimplePropertyTemplate; IfcTypeObject ; IfcTypeProcess ; IfcTypeProduct ; IfcTypeResource ; IfcPropertySetDefinitionSet; IfcComplexPropertyTemplateTypeEnum; IfcObjectTypeEnum; IfcPropertySetTemplateTypeEnum; IfcSimplePropertyTemplateTypeEnum;	42/60 (70%)

Namespace	Definitions	Usage
	IfcDefinitionSelect; IfcProcessSelect; IfcProductSelect; IfcPropertySetDefinitionSelect; IfcResourceSelect;	
IfcControlExtension	IfcPerformanceHistory; IfcRelAssociatesApproval; IfcRelAssociatesConstraint; IfcPerformanceHistoryTypeEnum;	1/4 (25%)
IfcProcessExtension	IfcEvent; IfcEventType; IfcProcedure; IfcProcedureType; IfcRelSequence; IfcTask; IfcTaskType; IfcWorkCalendar; IfcWorkControl; IfcWorkPlan; IfcWorkSchedule; IfcEventTriggerTypeEnum; IfcEventTypeEnum; IfcProcedureTypeEnum; IfcSequenceEnum; IfcTaskTypeEnum; IfcWorkCalendarTypeEnum; IfcWorkPlanTypeEnum; IfcWorkScheduleTypeEnum;	7/19 (37%)
IfcProductExtension	IfcAnnotation; IfcBuilding; IfcBuildingElement; IfcBuildingElementType; IfcBuildingStorey; IfcCivilElement; IfcCivilElementType; IfcDistributionElement; IfcDistributionElementType; IfcElement; IfcElementAssembly; IfcElementAssemblyType; IfcElementQuantity; IfcElementType; IfcExternalSpatialElement; IfcExternalSpatialStructureElement; IfcFeatureElement; IfcFeatureElementAddition; IfcFeatureElementSubtraction; IfcFurnishingElement; IfcFurnishingElementType; IfcGeographicElement; IfcGeographicElementType; IfcGrid; IfcOpeningElement; IfcOpeningStandardCase; IfcPort; IfcProjectionElement; IfcRelAssociatesMaterial; IfcRelConnectsElements; IfcRelConnectsPorts; IfcRelConnectsPortToElement; IfcRelConnectsWithRealizingElements; IfcRelContainedInSpatialStructure; IfcRelFillsElement; IfcRelInterferesElements; IfcRelProjectsElement; IfcRelReferencedInSpatialStructure; IfcRelServicesBuildings; IfcRelSpaceBoundary; IfcRelSpaceBoundary1stLevel; IfcRelSpaceBoundary2ndLevel; IfcRelVoidsElement; IfcSite; IfcSpace; IfcSpaceType; IfcSpatialElement; IfcSpatialElementType; IfcSpatialStructureElement; IfcSpatialStructureElementType; IfcSpatialZone; IfcSpatialZoneType; IfcSystem; IfcTransportElement; IfcTransportElementType; IfcVirtualElement; IfcZone; IfcAssemblyPlaceEnum; IfcElementAssemblyTypeEnum; IfcElementCompositionEnum; IfcExternalSpatialElementTypeEnum; IfcGeographicElementTypeEnum; IfcGridTypeEnum; IfcInternalOrExternalEnum; IfcOpeningElementTypeEnum; IfcPhysicalOrVirtualEnum; IfcProjectionElementTypeEnum; IfcSpaceTypeEnum; IfcSpatialZoneTypeEnum; IfcTransportElementTypeEnum; IfcSpaceBoundarySelect;	26/71 (37%)
IfcSharedBldgElements	IfcBeam; IfcBeamStandardCase; IfcBeamType; IfcBuildingElementProxy; IfcBuildingElementProxyType; IfcBuildingSystem; IfcChimney; IfcChimneyType; IfcColumn; IfcColumnStandardCase; IfcColumnType; IfcCovering; IfcCoveringType; IfcCurtainWall; IfcCurtainWallType; IfcDoor; IfcDoorStandardCase; IfcDoorType; IfcMember; IfcMemberStandardCase; IfcMemberType; IfcPlate; IfcPlateStandardCase; IfcPlateType; IfcRailing; IfcRailingType; IfcRamp; IfcRampFlight; IfcRampFlightType; IfcRampType; IfcRelConnectsPathElements; IfcRelCoversBldgElements; IfcRelCoversSpaces; IfcRoof; IfcRoofType; IfcShadingDevice; IfcShadingDeviceType; IfcSlab; IfcSlabElementedCase; IfcSlabStandardCase; IfcSlabType; IfcStair; IfcStairFlight; IfcStairFlightType; IfcStairType; IfcWall; IfcWallElementedCase; IfcWallStandardCase; IfcWallType; IfcWindow; IfcWindowStandardCase; IfcWindowType; IfcBeamTypeEnum; IfcBuildingElementProxyTypeEnum; IfcBuildingSystemTypeEnum; IfcChimneyTypeEnum; IfcColumnTypeEnum; IfcConnectionTypeEnum; IfcCoveringTypeEnum; IfcCurtainWallTypeEnum; IfcDoorTypeEnum; IfcDoorTypeEnumOperationEnum; IfcMemberTypeEnum; IfcPlateTypeEnum; IfcRailingTypeEnum; IfcRampFlightTypeEnum; IfcRampTypeEnum; IfcRoofTypeEnum; IfcShadingDeviceTypeEnum; IfcSlabTypeEnum; IfcStairFlightTypeEnum; IfcStairTypeEnum; IfcWallTypeEnum; IfcWindowTypeEnum; IfcWindowTypePartitioningEnum;	10/75 (13%)
IfcSharedBldgServiceElements	IfcDistributionChamberElement; IfcDistributionChamberElementType; IfcDistributionCircuit; IfcDistributionControlElement; IfcDistributionControlElementType; IfcDistributionFlowElement; IfcDistributionFlowElementType; IfcDistributionPort; IfcDistributionSystem; IfcEnergyConversionDevice; IfcEnergyConversionDeviceType; IfcFlowController; IfcFlowControllerType; IfcFlowFitting; IfcFlowFittingType; IfcFlowMovingDevice; IfcFlowMovingDeviceType; IfcFlowSegment; IfcFlowSegmentType; IfcFlowStorageDevice; IfcFlowStorageDeviceType; IfcFlowTerminal; IfcFlowTerminalType; IfcFlowTreatmentDevice; IfcFlowTreatmentDeviceType; IfcRelFlowControlElements; IfcDistributionChamberElementTypeEnum; IfcDistributionPortTypeEnum; IfcDistributionSystemEnum; IfcFlowDirectionEnum;	22/30 (73%)
IfcSharedComponentElements	IfcBuildingElementPart; IfcBuildingElementPartType; IfcDiscreteAccessory; IfcDiscreteAccessoryType; IfcElementComponent; IfcElementComponentType; IfcFastener;	2/14 (14%)

Namespace	Definitions	Usage
	IfcFastenerType; IfcMechanicalFastener; IfcMechanicalFastenerType; IfcBuildingElementPartTypeEnum; IfcDiscreteAccessoryTypeEnum; IfcFastenerTypeEnum; IfcMechanicalFastenerTypeEnum;	
IfcSharedFacilitiesElements	IfcAsset; IfcFurniture; IfcFurnitureType; IfcInventory; IfcOccupant ; IfcSystemFurnitureElement; IfcSystemFurnitureElementType; IfcFurnitureTypeEnum; IfcInventoryTypeEnum; IfcOccupantTypeEnum; IfcSystemFurnitureElementTypeEnum;	1/11 (9%)
IfcSharedMgmtElements	IfcActionRequest; IfcCostItem ; IfcCostSchedule ; IfcPermit; IfcProjectOrder; IfcActionRequestTypeEnum; IfcCostItemTypeEnum; IfcCostScheduleTypeEnum; IfcPermitTypeEnum; IfcProjectOrderTypeEnum;	2/10 (20%)
IfcArchitectureDomain	IfcDoorLiningProperties; IfcDoorPanelProperties; IfcDoorStyle; IfcPermeableCoveringProperties; IfcWindowLiningProperties; IfcWindowPanelProperties; IfcWindowStyle; IfcDoorPanelOperationEnum; IfcDoorPanelPositionEnum; IfcDoorStyleConstructionEnum; IfcDoorStyleOperationEnum; IfcPermeableCoveringOperationEnum; IfcWindowPanelOperationEnum; IfcWindowPanelPositionEnum; IfcWindowStyleConstructionEnum; IfcWindowStyleOperationEnum;	0/16 (0%)
IfcBuildingControlsDomain	IfcActuator; IfcActuatorType; IfcAlarm; IfcAlarmType; IfcController; IfcControllerType; IfcFlowInstrument; IfcFlowInstrumentType; IfcSensor; IfcSensorType; IfcUnitaryControlElement; IfcUnitaryControlElementType; IfcActuatorTypeEnum; IfcAlarmTypeEnum; IfcControllerTypeEnum; IfcFlowInstrumentTypeEnum; IfcSensorTypeEnum; IfcUnitaryControlElementTypeEnum;	0/18 (0%)
IfcConstructionMgmtDomain	IfcConstructionEquipmentResource ; IfcConstructionEquipmentResourceType ; IfcConstructionMaterialResource; IfcConstructionMaterialResourceType; IfcConstructionProductResource; IfcConstructionProductResourceType; IfcConstructionResource ; IfcConstructionResourceType ; IfcCrewResource; IfcCrewResourceType; IfcLaborResource; IfcLaborResourceType; IfcSubContractResource ; IfcSubContractResourceType ; IfcConstructionEquipmentResourceTypeEnum ; IfcConstructionMaterialResourceTypeEnum; IfcConstructionProductResourceTypeEnum; IfcCrewResourceTypeEnum; IfcLaborResourceTypeEnum; IfcSubContractResourceTypeEnum ;	8/20 (40%)
IfcElectricalDomain	IfcAudioVisualAppliance; IfcAudioVisualApplianceType; IfcCableCarrierFitting ; IfcCableCarrierFittingType ; IfcCableCarrierSegment ; IfcCableCarrierSegmentType ; IfcCableFitting ; IfcCableFittingType ; IfcCableSegment ; IfcCableSegmentType ; IfcCommunicationsAppliance; IfcCommunicationsApplianceType; IfcElectricAppliance ; IfcElectricApplianceType ; IfcElectricDistributionBoard ; IfcElectricDistributionBoardType ; IfcElectricFlowStorageDevice ; IfcElectricFlowStorageDeviceType ; IfcElectricGenerator ; IfcElectricGeneratorType ; IfcElectricMotor ; IfcElectricMotorType ; IfcElectricTimeControl; IfcElectricTimeControlType; IfcJunctionBox ; IfcJunctionBoxType ; IfcLamp ; IfcLampType ; IfcLightFixture ; IfcLightFixtureType ; IfcMotorConnection; IfcMotorConnectionType; IfcOutlet ; IfcOutletType ; IfcProtectiveDevice ; IfcProtectiveDeviceTrippingUnit; IfcProtectiveDeviceTrippingUnitType; IfcProtectiveDeviceType ; IfcSolarDevice ; IfcSolarDeviceType ; IfcSwitchingDevice ; IfcSwitchingDeviceType ; IfcTransformer ; IfcTransformerType ; IfcAudioVisualApplianceTypeEnum; IfcCableCarrierFittingTypeEnum ; IfcCableCarrierSegmentTypeEnum ; IfcCableFittingTypeEnum ; IfcCableSegmentTypeEnum ; IfcCommunicationsApplianceTypeEnum; IfcElectricApplianceTypeEnum ; IfcElectricDistributionBoardTypeEnum ; IfcElectricFlowStorageDeviceTypeEnum ; IfcElectricGeneratorTypeEnum ; IfcElectricMotorTypeEnum ; IfcElectricTimeControlTypeEnum; IfcJunctionBoxTypeEnum ; IfcLampTypeEnum ; IfcLightFixtureTypeEnum ; IfcMotorConnectionTypeEnum; IfcOutletTypeEnum ; IfcProtectiveDeviceTrippingUnitTypeEnum; IfcProtectiveDeviceTypeEnum ; IfcSolarDeviceTypeEnum ; IfcSwitchingDeviceTypeEnum ; IfcTransformerTypeEnum ;	51/66 (77%)
IfcHvacDomain	IfcAirTerminal; IfcAirTerminalBox; IfcAirTerminalBoxType; IfcAirTerminalType; IfcAirToAirHeatRecovery; IfcAirToAirHeatRecoveryType; IfcBoiler; IfcBoilerType; IfcBurner; IfcBurnerType; IfcChiller ; IfcChillerType ; IfcCoil; IfcCoilType; IfcCompressor; IfcCompressorType; IfcCondenser; IfcCondenserType; IfcCooledBeam; IfcCooledBeamType; IfcCoolingTower; IfcCoolingTowerType; IfcDamper; IfcDamperType; IfcDuctFitting; IfcDuctFittingType; IfcDuctSegment; IfcDuctSegmentType; IfcDuctSilencer; IfcDuctSilencerType; IfcEngine; IfcEngineType; IfcEvaporativeCooler; IfcEvaporativeCoolerType; IfcEvaporator; IfcEvaporatorType; IfcFan; IfcFanType; IfcFilter; IfcFilterType; IfcFlowMeter ; IfcFlowMeterType ; IfcHeatExchanger; IfcHeatExchangerType;	12/99 (12%)

Namespace	Definitions	Usage
	IfcHumidifier; IfcHumidifierType; IfcMedicalDevice; IfcMedicalDeviceType; IfcPipeFitting; IfcPipeFittingType; IfcPipeSegment; IfcPipeSegmentType; IfcPump; IfcPumpType; IfcSpaceHeater; IfcSpaceHeaterType; IfcTank; IfcTankType; IfcTubeBundle; IfcTubeBundleType; IfcUnitaryEquipment; IfcUnitaryEquipmentType; IfcValve; IfcValveType; IfcVibrationIsolator; IfcVibrationIsolatorType; IfcAirTerminalBoxTypeEnum; IfcAirTerminalTypeEnum; IfcAirToAirHeatRecoveryTypeEnum; IfcBoilerTypeEnum; IfcBurnerTypeEnum; IfcChillerTypeEnum; IfcCoilTypeEnum; IfcCompressorTypeEnum; IfcCondenserTypeEnum; IfcCooledBeamTypeEnum; IfcCoolingTowerTypeEnum; IfcDuctFittingTypeEnum; IfcDuctSegmentTypeEnum; IfcDuctSilencerTypeEnum; IfcEngineTypeEnum; IfcEvaporativeCoolerTypeEnum; IfcEvaporatorTypeEnum; IfcFanTypeEnum; IfcFilterTypeEnum; IfcFlowMeterTypeEnum; IfcHeatExchangerTypeEnum; IfcHumidifierTypeEnum; IfcMedicalDeviceTypeEnum; IfcPipeFittingTypeEnum; IfcPipeSegmentTypeEnum; IfcPumpTypeEnum; IfcSpaceHeaterTypeEnum; IfcTankTypeEnum; IfcTubeBundleTypeEnum; IfcUnitaryEquipmentTypeEnum; IfcValveTypeEnum; IfcVibrationIsolatorTypeEnum;	
IfcPlumbingFireProtectionDomain	IfcFireSuppressionTerminal; IfcFireSuppressionTerminalType; IfcInterceptor; IfcInterceptorType; IfcSanitaryTerminal; IfcSanitaryTerminalType; IfcStackTerminal; IfcStackTerminalType; IfcWasteTerminal; IfcWasteTerminalType; IfcFireSuppressionTerminalTypeEnum; IfcInterceptorTypeEnum; IfcSanitaryTerminalTypeEnum; IfcStackTerminalTypeEnum; IfcWasteTerminalTypeEnum;	0/15 (0%)
IfcStructuralAnalysisDomain	IfcRelConnectsStructuralActivity; IfcRelConnectsStructuralMember; IfcRelConnectsWithEccentricity; IfcStructuralAction; IfcStructuralActivity; IfcStructuralAnalysisModel; IfcStructuralConnection; IfcStructuralCurveAction; IfcStructuralCurveConnection; IfcStructuralCurveMember; IfcStructuralCurveMemberVarying; IfcStructuralCurveReaction; IfcStructuralItem; IfcStructuralLinearAction; IfcStructuralLoadCase; IfcStructuralLoadGroup; IfcStructuralMember; IfcStructuralPlanarAction; IfcStructuralPointAction; IfcStructuralPointConnection; IfcStructuralPointReaction; IfcStructuralReaction; IfcStructuralResultGroup; IfcStructuralSurfaceAction; IfcStructuralSurfaceConnection; IfcStructuralSurfaceMember; IfcStructuralSurfaceMemberVarying; IfcStructuralSurfaceReaction; IfcActionSourceTypeEnum; IfcActionTypeEnum; IfcAnalysisModelTypeEnum; IfcAnalysisTheoryTypeEnum; IfcLoadGroupTypeEnum; IfcProjectedOrTrueLengthEnum; IfcStructuralCurveActivityTypeEnum; IfcStructuralCurveMemberTypeEnum; IfcStructuralSurfaceActivityTypeEnum; IfcStructuralSurfaceMemberTypeEnum; IfcStructuralActivityAssignmentSelect;	0/39 (0%)
IfcStructuralElementsDomain	IfcFooting; IfcFootingType; IfcPile; IfcPileType; IfcReinforcementDefinitionProperties; IfcReinforcingBar; IfcReinforcingBarType; IfcReinforcingElement; IfcReinforcingElementType; IfcReinforcingMesh; IfcReinforcingMeshType; IfcSurfaceFeature; IfcTendon; IfcTendonAnchor; IfcTendonAnchorType; IfcTendonType; IfcVoidingFeature; IfcFootingTypeEnum; IfcPileConstructionEnum; IfcPileTypeEnum; IfcReinforcingBarTypeEnum; IfcReinforcingMeshTypeEnum; IfcSurfaceFeatureTypeEnum; IfcTendonAnchorTypeEnum; IfcTendonTypeEnum; IfcVoidingFeatureTypeEnum; IfcBendingParameterSelect;	0/27 (0%)
IfcActorResource	IfcActorRole; IfcAddress; IfcOrganization; IfcOrganizationRelationship; IfcPerson; IfcPersonAndOrganization; IfcPostalAddress; IfcTelecomAddress; IfcAddressTypeEnum; IfcRoleEnum; IfcActorSelect;	9/11 (82%)
IfcApprovalResource	IfcApproval; IfcApprovalRelationship; IfcResourceApprovalRelationship;	0/3 (0%)
IfcConstraintResource	IfcConstraint; IfcMetric; IfcObjective; IfcReference; IfcResourceConstraintRelationship; IfcBenchmarkEnum; IfcConstraintEnum; IfcLogicalOperatorEnum; IfcObjectiveEnum; IfcMetricValueSelect;	0/10 (0%)
IfcCostResource	IfcAppliedValue; IfcCostValue; IfcCurrencyRelationship; IfcArithmeticOperatorEnum; IfcAppliedValueSelect;	3/5 (60%)
IfcDateTimeResource	IfcEventTime; IfcIrregularTimeSeries; IfcIrregularTimeSeriesValue; IfcLagTime; IfcRecurrencePattern; IfcRegularTimeSeries; IfcResourceTime; IfcSchedulingTime; IfcTaskTime; IfcTaskTimeRecurring; IfcTimePeriod; IfcTimeSeries; IfcTimeSeriesValue; IfcWorkTime; IfcDate; IfcDateTime; IfcDayInMonthNumber; IfcDayInWeekNumber; IfcDuration; IfcMonthInYearNumber; IfcTime; IfcTimeStamp; IfcDataOriginEnum; IfcRecurrenceTypeEnum; IfcTaskDurationEnum; IfcTimeSeriesDataTypeEnum; IfcTimeOrRatioSelect;	20/27 (74%)
IfcExternalReferenceResource	IfcClassification; IfcClassificationReference; IfcDocumentInformation; IfcDocumentInformationRelationship; IfcDocumentReference; IfcExternalInformation;	6/20 (30%)

Namespace	Definitions	Usage
	IfcExternalReference ; IfcExternalReferenceRelationship; IfcLibraryInformation; IfcLibraryReference; IfcResourceLevelRelationship; IfcLanguageId; IfcURIReference; IfcDocumentConfidentialityEnum; IfcDocumentStatusEnum; IfcClassificationReferenceSelect ; IfcClassificationSelect ; IfcDocumentSelect; IfcLibrarySelect; IfcResourceObjectSelect;	
IfcGeometricConstraintResource	IfcConnectionCurveGeometry; IfcConnectionGeometry; IfcConnectionPointEccentricity; IfcConnectionPointGeometry; IfcConnectionSurfaceGeometry; IfcConnectionVolumeGeometry; IfcGridAxis ; IfcGridPlacement ; IfcLocalPlacement ; IfcObjectPlacement ; IfcVirtualGridIntersection ; IfcCurveOrEdgeCurve; IfcGridPlacementDirectionSelect; IfcPointOrVertexPoint; IfcSolidOrShell; IfcSurfaceOrFaceSurface;	5/16 (31%)
IfcGeometricModelResource	IfcAdvancedBrep ; IfcAdvancedBrepWithVoids; IfcBlock; IfcBooleanClippingResult ; IfcBooleanResult ; IfcBoundingBox; IfcBoxedHalfSpace; IfcCartesianPointList ; IfcCartesianPointList3D ; IfcCsgPrimitive3D; IfcCsgSolid ; IfcExtrudedAreaSolid ; IfcExtrudedAreaSolidTapered; IfcFaceBasedSurfaceModel ; IfcFacetedBrep ; IfcFacetedBrepWithVoids; IfcFixedReferenceSweptAreaSolid; IfcGeometricCurveSet ; IfcGeometricSet ; IfcHalfSpaceSolid; IfcManifoldSolidBrep ; IfcPolygonalBoundedHalfSpace; IfcRectangularPyramid; IfcRevolvedAreaSolid ; IfcRevolvedAreaSolidTapered; IfcRightCircularCone; IfcRightCircularCylinder; IfcSectionedSpine ; IfcShellBasedSurfaceModel ; IfcSolidModel ; IfcSphere; IfcSurfaceCurveSweptAreaSolid; IfcSweptAreaSolid ; IfcSweptDiskSolid ; IfcSweptDiskSolidPolygonal; IfcTessellatedFaceSet ; IfcTessellatedItem ; IfcTriangulatedFaceSet ; IfcBooleanOperator ; IfcBooleanOperand ; IfcCsgSelect ; IfcGeometricSetSelect ;	25/42 (60%)
IfcGeometryResource	IfcAxis1Placement ; IfcAxis2Placement2D; IfcAxis2Placement3D ; IfcBoundaryCurve; IfcBoundedCurve ; IfcBoundedSurface ; IfcBSplineCurve; IfcBSplineCurveWithKnots; IfcBSplineSurface; IfcBSplineSurfaceWithKnots; IfcCartesianPoint ; IfcCartesianTransformationOperator ; IfcCartesianTransformationOperator2D; IfcCartesianTransformationOperator2DnonUniform; IfcCartesianTransformationOperator3D; IfcCartesianTransformationOperator3DnonUniform; IfcCircle; IfcCompositeCurve ; IfcCompositeCurveOnSurface; IfcCompositeCurveSegment ; IfcConic; IfcCurve ; IfcCurveBoundedPlane; IfcCurveBoundedSurface; IfcCylindricalSurface; IfcDirection ; IfcElementarySurface; IfcEllipse; IfcGeometricRepresentationItem ; IfcLine ; IfcMappedItem ; IfcOffsetCurve2D; IfcOffsetCurve3D; IfcOuterBoundaryCurve; IfcPcurve; IfcPlacement ; IfcPlane; IfcPoint ; IfcPointOnCurve; IfcPointOnSurface; IfcPolyline; IfcRationalBSplineCurveWithKnots; IfcRationalBSplineSurfaceWithKnots; IfcRectangularTrimmedSurface; IfcReparametrisedCompositeCurveSegment; IfcRepresentationItem ; IfcRepresentationMap ; IfcSurface ; IfcSurfaceOfLinearExtrusion; IfcSurfaceOfRevolution; IfcSweptSurface ; IfcTrimmedCurve; IfcVector ; IfcDimensionCount ; IfcBSplineCurveForm; IfcBSplineSurfaceForm; IfcKnotType; IfcTransitionCode ; IfcTrimmingPreference; IfcAxis2Placement ; IfcCurveOnSurface; IfcTrimmingSelect; IfcVectorOrDirection;	23/63 (37%)
IfcMaterialResource	IfcMaterial ; IfcMaterialClassificationRelationship; IfcMaterialConstituent; IfcMaterialConstituentSet; IfcMaterialDefinition ; IfcMaterialLayer; IfcMaterialLayerSet; IfcMaterialLayerSetUsage; IfcMaterialLayerWithOffsets; IfcMaterialList; IfcMaterialProfile ; IfcMaterialProfileSet ; IfcMaterialProfileSetUsage ; IfcMaterialProfileSetUsageTapering; IfcMaterialProfileWithOffsets; IfcMaterialProperties ; IfcMaterialRelationship; IfcMaterialUsageDefinition ; IfcCardinalPointReference; IfcDirectionSenseEnum; IfcLayerSetDirectionEnum; IfcMaterialSelect ;	8/22 (36%)
IfcMeasureResource	IfcContextDependentUnit; IfcConversionBasedUnit ; IfcConversionBasedUnitWithOffset ; IfcDerivedUnitElement; IfcDimensionalExponents ; IfcMeasureWithUnit ; IfcMonetaryUnit; IfcNamedUnit ; IfcSIUnit ; IfcUnitAssignment ; IfcAbsorbedDoseMeasure; IfcAccelerationMeasure; IfcAmountOfSubstanceMeasure; IfcAngularVelocityMeasure; IfcAreaDensityMeasure; IfcAreaMeasure ; IfcBoolean ; IfcComplexNumber; IfcCompoundPlaneAngleMeasure ; IfcContextDependentMeasure; IfcCountMeasure ; IfcCurvatureMeasure; IfcDescriptiveMeasure; IfcDoseEquivalentMeasure; IfcDynamicViscosityMeasure; IfcElectricCapacitanceMeasure; IfcElectricChargeMeasure; IfcElectricConductanceMeasure; IfcElectricCurrentMeasure ; IfcElectricResistanceMeasure ; IfcElectricVoltageMeasure ; IfcEnergyMeasure ; IfcForceMeasure; IfcFrequencyMeasure ; IfcHeatFluxDensityMeasure; IfcHeatingValueMeasure; IfcIdentifier ; IfcIlluminanceMeasure ; IfcInductanceMeasure; IfcInteger ; IfcIntegerCountRateMeasure; IfcIonConcentrationMeasure; IfcIsothermalMoistureCapacityMeasure; IfcKinematicViscosityMeasure; IfcLabel ; IfcLengthMeasure ; IfcLinearForceMeasure; IfcLinearMomentMeasure; IfcLinearStiffnessMeasure; IfcLinearVelocityMeasure; IfcLogical ; IfcLuminousFluxMeasure;	41/121 (34%)

Namespace	Definitions	Usage
	<p>IfcLuminousIntensityDistributionMeasure; IfcLuminousIntensityMeasure; IfcMagneticFluxDensityMeasure; IfcMagneticFluxMeasure; IfcMassDensityMeasure; IfcMassFlowRateMeasure; IfcMassMeasure; IfcMassPerLengthMeasure; IfcModulusOfElasticityMeasure; IfcModulusOfLinearSubgradeReactionMeasure; IfcModulusOfRotationalSubgradeReactionMeasure; IfcModulusOfSubgradeReactionMeasure; IfcMoistureDiffusivityMeasure; IfcMolecularWeightMeasure; IfcMomentOfInertiaMeasure; IfcMonetaryMeasure; IfcNonNegativeLengthMeasure; IfcNormalisedRatioMeasure; IfcNumericMeasure; IfcParameterValue; IfcPHMeasure; IfcPlanarForceMeasure; IfcPlaneAngleMeasure; IfcPositiveLengthMeasure; IfcPositivePlaneAngleMeasure; IfcPositiveRatioMeasure; IfcPowerMeasure; IfcPressureMeasure; IfcRadioActivityMeasure; IfcRatioMeasure; IfcReal; IfcRotationalFrequencyMeasure; IfcRotationalMassMeasure; IfcRotationalStiffnessMeasure; IfcSectionalAreaIntegralMeasure; IfcSectionModulusMeasure; IfcShearModulusMeasure; IfcSolidAngleMeasure; IfcSoundPowerLevelMeasure; IfcSoundPowerMeasure; IfcSoundPressureLevelMeasure; IfcSoundPressureMeasure; IfcSpecificHeatCapacityMeasure; IfcTemperatureGradientMeasure; IfcTemperatureRateOfChangeMeasure; IfcText; IfcThermalAdmittanceMeasure; IfcThermalConductivityMeasure; IfcThermalExpansionCoefficientMeasure; IfcThermalResistanceMeasure; IfcThermalTransmittanceMeasure; IfcThermodynamicTemperatureMeasure; IfcTimeMeasure; IfcTorqueMeasure; IfcVaporPermeabilityMeasure; IfcVolumeMeasure; IfcVolumetricFlowRateMeasure; IfcWarpingMomentMeasure; IfcDerivedUnitEnum; IfcSI Prefix; IfcSI UnitName; IfcUnitEnum; IfcDerivedMeasureValue; IfcMeasureValue; IfcSimpleValue; IfcUnit; IfcValue;</p>	
IfcPresentationAppearanceResource	<p>IfcBlobTexture; IfcColourRgb; IfcColourRgbList; IfcColourSpecification; IfcCurveStyle; IfcCurveStyleFont; IfcCurveStyleFontAndScaling; IfcCurveStyleFontPattern; IfcDraughtingPreDefinedColour; IfcDraughtingPreDefinedCurveFont; IfcExternallyDefinedHatchStyle; IfcExternallyDefinedSurfaceStyle; IfcExternallyDefinedTextFont; IfcFillAreaStyle; IfcFillAreaStyleHatching; IfcFillAreaStyleTiles; IfcImageTexture; IfcIndexedColourMap; IfcIndexedTextureMap; IfcIndexedTriangleTextureMap; IfcPixelTexture; IfcPreDefinedColour; IfcPreDefinedCurveFont; IfcPreDefinedItem; IfcPreDefinedTextFont; IfcPresentationStyle; IfcPresentationStyleAssignment; IfcStyledItem; IfcSurfaceStyle; IfcSurfaceStyleLighting; IfcSurfaceStyleRefraction; IfcSurfaceStyleRendering; IfcSurfaceStyleShading; IfcSurfaceStyleWithTextures; IfcSurfaceTexture; IfcTextStyle; IfcTextStyleFontModel; IfcTextStyleForDefinedFont; IfcTextStyleTextModel; IfcTextureCoordinate; IfcTextureCoordinateGenerator; IfcTextureMap; IfcTextureVertex; IfcTextureVertexList; IfcFontStyle; IfcFontVariant; IfcFontWeight; IfcPresentableText; IfcSpecularExponent; IfcSpecularRoughness; IfcTextAlignment; IfcTextDecoration; IfcTextFontName; IfcTextTransformation; IfcNullStyle; IfcReflectanceMethodEnum; IfcSurfaceSide; IfcColour; IfcColourOrFactor; IfcCurveFontOrScaledCurveFontSelect; IfcCurveStyleFontSelect; IfcFillStyleSelect; IfcHatchLineDistanceSelect; IfcPresentationStyleSelect; IfcSizeSelect; IfcSpecularHighlightSelect; IfcStyleAssignmentSelect; IfcSurfaceStyleElementSelect; IfcTextFontSelect;</p>	12/69 (17%)
IfcPresentationDefinitionResource	<p>IfcAnnotationFillArea; IfcPlanarBox; IfcPlanarExtent; IfcPresentationItem; IfcTextLiteral; IfcTextLiteralWithExtent; IfcBoxAlignment; IfcTextPath;</p>	4/8 (50%)
IfcPresentationOrganizationResource	<p>IfcLightDistributionData; IfcLightIntensityDistribution; IfcLightSource; IfcLightSourceAmbient; IfcLightSourceDirectional; IfcLightSourceGoniometric; IfcLightSourcePositional; IfcLightSourceSpot; IfcPresentationLayerAssignment; IfcPresentationLayerWithStyle; IfcLightDistributionCurveEnum; IfcLightEmissionSourceEnum; IfcLayeredItem; IfcLightDistributionDataSourceSelect;</p>	3/14 (21%)
IfcProfileResource	<p>IfcArbitraryClosedProfileDef; IfcArbitraryOpenProfileDef; IfcArbitraryProfileDefWithVoids; IfcAsymmetricIShapeProfileDef; IfcCenterLineProfileDef; IfcCircleHollowProfileDef; IfcCircleProfileDef; IfcCompositeProfileDef; IfcCShapeProfileDef; IfcDerivedProfileDef; IfcEllipseProfileDef; IfcIShapeProfileDef; IfcLShapeProfileDef; IfcMirroredProfileDef; IfcParameterizedProfileDef; IfcProfileDef; IfcProfileProperties; IfcRectangleHollowProfileDef; IfcRectangleProfileDef; IfcReinforcementBarProperties; IfcRoundedRectangleProfileDef; IfcSectionProperties; IfcSectionReinforcementProperties; IfcTrapeziumProfileDef; IfcTShapeProfileDef; IfcUShapeProfileDef; IfcZShapeProfileDef; IfcProfileTypeEnum; IfcReinforcingBarRoleEnum; IfcReinforcingBarSurfaceEnum; IfcSectionTypeEnum;</p>	3/31 (10%)
IfcPropertyResource	<p>IfcComplexProperty; IfcExtendedProperties; IfcPreDefinedProperties; IfcProperty; IfcPropertyAbstraction; IfcPropertyBoundedValue; IfcPropertyDependencyRelationship;</p>	11/16 (69%)

Namespace	Definitions	Usage
	IfcPropertyEnumeratedValue ; IfcPropertyEnumeration; IfcPropertyListValue ; IfcPropertyReferenceValue ; IfcPropertySingleValue ; IfcPropertyTableValue ; IfcSimpleProperty ; IfcCurveInterpolationEnum; IfcObjectReferenceSelect ;	
IfcQuantityResource	IfcPhysicalComplexQuantity; IfcPhysicalQuantity ; IfcPhysicalSimpleQuantity ; IfcQuantityArea; IfcQuantityCount; IfcQuantityLength; IfcQuantityTime; IfcQuantityVolume; IfcQuantityWeight;	2/9 (22%)
IfcRepresentationResource	IfcCoordinateOperation; IfcCoordinateReferenceSystem; IfcGeometricRepresentationContext ; IfcGeometricRepresentationSubContext; IfcMapConversion; IfcMaterialDefinitionRepresentation ; IfcProductDefinitionShape ; IfcProductRepresentation ; IfcProjectedCRS; IfcRepresentation ; IfcRepresentationContext ; IfcShapeAspect; IfcShapeModel ; IfcShapeRepresentation ; IfcStyledRepresentation ; IfcStyleModel ; IfcTopologyRepresentation; IfcGeometricProjectionEnum; IfcGlobalOrLocalEnum; IfcCoordinateReferenceSystemSelect; IfcProductRepresentationSelect;	10/21 (48%)
IfcStructuralLoadResource	IfcBoundaryCondition; IfcBoundaryEdgeCondition; IfcBoundaryFaceCondition; IfcBoundaryNodeCondition; IfcBoundaryNodeConditionWarping; IfcFailureConnectionCondition; IfcSlippageConnectionCondition; IfcStructuralConnectionCondition; IfcStructuralLoad; IfcStructuralLoadConfiguration; IfcStructuralLoadLinearForce; IfcStructuralLoadOrResult; IfcStructuralLoadPlanarForce; IfcStructuralLoadSingleDisplacement; IfcStructuralLoadSingleDisplacementDistortion; IfcStructuralLoadSingleForce; IfcStructuralLoadSingleForceWarping; IfcStructuralLoadStatic; IfcStructuralLoadTemperature; IfcSurfaceReinforcementArea; IfcModulusOfRotationalSubgradeReactionSelect; IfcModulusOfSubgradeReactionSelect; IfcModulusOfTranslationalSubgradeReactionSelect; IfcRotationalStiffnessSelect; IfcTranslationalStiffnessSelect; IfcWarpingStiffnessSelect;	0/26 (0%)
IfcTopologyResource	IfcAdvancedFace; IfcClosedShell ; IfcConnectedFaceSet ; IfcEdge; IfcEdgeCurve; IfcEdgeLoop; IfcFace ; IfcFaceBound ; IfcFaceOuterBound; IfcFaceSurface; IfcLoop ; IfcOpenShell; IfcOrientedEdge; IfcPath; IfcPolyLoop; IfcSubedge; IfcTopologicalRepresentationItem ; IfcVertex; IfcVertexLoop; IfcVertexPoint; IfcShell ;	7/21 (33%)
IfcUtilityResource	IfcApplication ; IfcOwnerHistory ; IfcTable; IfcTableColumn; IfcTableRow; IfcGloballyUniqueId ; IfcChangeActionEnum ; IfcStateEnum ;	5/8 (63%)

4.7.7 Conformance Testing Procedures

4.7.7.1 Format and content requirements

4.7.7.1.1 Test rule list

The list of test rules is defined in the mvdXML file referenced herein.

4.7.7.1.2. Test rule definition

The definitions of test rules are included in the mvdXML file referenced herein.

4.7.7.1.3 Test rules formatting

Formatting documentation for MVDXML is available at <http://www.buildingsmart-tech.org/specifications/mvd-overview/mvd-overview-summary>.

4.7.7.1.4 Test rule coverage analysis

Coverage of test rules for a given IFC file may be evaluated by using the mvdXML file referenced herein with the IfcDoc tool available at <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-beta-summary>.

4.7.7.2 Examples and mapping requirements

4.7.7.2.1 Example file list

The list of example files is available at http://www.nibs.org/?page=bsa_commonbimfiles.

4.7.7.2.2 Example file description

File descriptions are available at the website identified.

4.7.7.2.3 Common BIM file reuse

Common BIM files are re-used at the website identified.

4.7.7.2.4 Implementers' agreements

Implementers agreements are available at <http://www.buildingsmart-tech.org/implementation/ifc-implementation/ifc-impl-agreements/ifc-impl-agreements-summary>.

4.7.7.2.5 Transformations/mapping allowed

Transformations are defined in the MVDXML file referenced herein.

4.7.7.2.6 Transformation/mapping documentation

Transformation documentation for MVDXML is available at <http://www.buildingsmart-tech.org/specifications/mvd-overview/mvd-overview-summary>.

4.7.7.3 Testing tools and procedures

4.7.7.3.1 Testing tool list

IFCDOC is a Windows application that provides functionality for validating files against model view definitions, as well as authoring model view definitions. It is published by BuildingSMART International Ltd and is freely available at <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-beta-summary>

4.7.7.3.2 Testing tool algorithm

The algorithm for testing files is shown in the following C# source code for IFCDOC.

4.7.7.3.2.1 Model view validation algorithm

The core algorithm iterates through selected model views, iterates through concept roots (applying to an entity), finds all instances of the applicable entity, iterates through concepts on each entity, and validates each concept.

```
// iterate through each concept root
foreach (DocModelView docView in this.m_project.ModelViews)
{
    if (docView.Visible)
    {
        foreach (DocConceptRoot docRoot in docView.ConceptRoots)
```

```

{
    Type typeEntity = null;
    if (typemap.TryGetValue(docRoot.ApplicableEntity.Name.ToUpper(), out typeEntity))
    {
        // build list of instances
        List<SEntity> list = new List<SEntity>();
        foreach (SEntity instance in format.Instances.Values)
        {
            if (typeEntity.IsInstanceOfType(instance))
            {
                list.Add(instance);
            }
        }

        foreach (DocTemplateUsage docUsage in docRoot.Concepts)
        {
            bool eachresult = true; // assume passing unless something fails

            // if no template parameters defined, then evaluate generically
            if (docUsage.Items.Count == 0)
            {
                int fail = 0;
                int pass = 0;
                foreach (SEntity ent in list)
                {
                    // check with parameters plugged in
                    bool? result = true;
                    foreach (DocModelRule rule in docUsage.Definition.Rules)
                    {
                        result = rule.Validate(ent, null, typemap);
                        if (result != null && !result.Value)
                            break;
                    }

                    if (result == null)
                    {
                        // no applicable rules, so passing
                        pass++;
                    }
                    else if (result != null && result.Value)
                    {
                        // all rules passed
                        pass++;
                    }
                    else
                    {
                        fail++;
                    }
                }
            }

            foreach (DocTemplateItem docItem in docUsage.Items)
            {
                int pass = 0;
                int fail = 0;
                foreach (SEntity ent in list)
                {
                    // check with parameters plugged in
                    bool? result = true;
                    foreach (DocModelRule rule in docUsage.Definition.Rules)
                    {
                        result = rule.Validate(ent, docItem, typemap);
                        if (result != null && !result.Value)
                            break;
                    }
                }
            }
        }
    }
}

```

```
}  
  
if (result == null)  
{  
    // inapplicable; passes  
    pass++;  
}  
else if (result != null && result.Value)  
{  
    // applicable and valid; passes  
    pass++;  
}  
else  
{  
    fail++;  
}  
  
}  
  
}  
  
}  
  
}
```

4.7.7.3.2.2 Attribute rule validation algorithm

These support routines validate attribute rules.

```

/// <summary>
/// Validates an object to meet rule.
/// </summary>
/// <param name="target">Required instance to validate.</param>
/// <param name="docItem">Optional template parameters to use for validation.</param>
/// <param name="typemap">Map of types to resolve.</param>
/// <returns></returns>
public override bool? Validate(object target, DocTemplateItem docItem, Dictionary<string, Type> typemap)
{
    if (target == null)
        return false;

    // (1) check if field is defined on target object; if not, then this rule does not apply.
    FieldInfo fieldInfo = target.GetType().GetField(this.Name);
    if (fieldInfo == null)
        return false;

    // (2) extract the value
    object value = fieldInfo.GetValue(target); // may be null

    if (value is System.Collections.IList)
    {
        System.Collections.IList list = (System.Collections.IList)value;
        int pass = 0;
        int fail = 0;
        foreach (object o in list)
        {
            bool? result = ValidateItem(o, docItem, typemap);
            if (result != null)
            {
                if (result.Value)
                {
                    pass++;
                }
                else
                {
                    fail++;
                }
            }
        }
        return pass > fail ? true : false;
    }
    return result;
}

```



```

        {
            fail++;
        }
    }
}

if (this.CardinalityMin == 0 && this.CardinalityMax == 0)
{
    return (pass == 0);
}
else if (this.CardinalityMin == 0 && this.CardinalityMax == 1)
{
    return (pass == 0 || pass == 1);
}
else if (this.CardinalityMin == 1 && this.CardinalityMax == 1)
{
    return (pass == 1);
}
else if (this.CardinalityMin == 1)
{
    return (fail == 0);
}
else
{
    return true;
}
}
else
{
    // validate single
    return ValidateItem(value, docItem, typemap);
}
}

/// <summary>
/// Checks a value to see if it matches the parameter value.
/// </summary>
/// <param name="value"></param>
/// <param name="docItem"></param>
/// <param name="typemap"></param>
/// <returns>True if passing, False if failing, or Null if inapplicable.</returns>
private bool? ValidateItem(object value, DocTemplateItem docItem, Dictionary<string, Type> typemap)
{
    // (3) if parameter is defined, check for match
    if (!String.IsNullOrEmpty(this.Identification))
    {
        if (docItem == null)
            return true; // parameter must be specified in order to check this rule

        string match = docItem.GetParameterValue(this.Identification);
        if (value == null && String.IsNullOrEmpty(match))
        {
            return true;
        }
        else if (value is SEntity)
        {
            if (match != null && value.GetType().Name.Equals(match))
            {
                return true;
            }
        }
        else
        {
            return false;
        }
    }
}

```

```

    }
}
else if (value != null)
{
    // pull out internal value type
    FieldInfo fieldinfo = value.GetType().GetField("Value");
    if (fieldinfo != null)
    {
        object innervalue = fieldinfo.GetValue(value);
        if (innervalue == null)
        {
            return false;
        }
        else if (match != null && innervalue.ToString().Equals(match.ToString(), StringComparison.Ordinal))
        {
            return true;
        }
        else if (this.IsCondition())
        {
            // condition didn't match, so chain of rules does not apply -- return null.
            return null;
        }
        else
        {
            // constraint evaluated to false and conditioned applied.
            return false;
        }
    }
}
else
{
    return false;
}
}
else
{
    return false;
}
}

// (4) recurse through constraints or entity rules
if (this.Rules != null && this.Rules.Count > 0)
{
    foreach (DocModelRule rule in this.Rules)
    {
        // attribute rule is true if at least one entity filter matches or one constraint filter matches
        bool? result = rule.Validate(value, docItem, typemap);
        if (result != null && result.Value)
            return result;
    }

    return false;
}

return true;
}

```

4.7.7.3.2.3 Entity rule validation algorithm

These support routines validate entity rules.

```

/// <summary>
/// Validates rules for an entity.
/// </summary>

```

```

/// <param name="target">Required object to validate.</param>
/// <param name="docItem">Template item to validate.</param>
/// <param name="typemap">Map of type names to type definitions.</param>
/// <returns>True if passing, False if failing, or Null if inapplicable.</returns>
public override bool? Validate(object target, DocTemplateItem docItem, Dictionary<string, Type> typemap)
{
    // checking for matching cast
    Type t = null;
    if (!typemap.TryGetValue(this.Name.ToUpper(), out t))
        return false;

    if (!t.IsInstanceOfType(target))
        return false;

    if (target is SEntity)
    {
        foreach (DocModelRule rule in this.Rules)
        {
            bool? result = rule.Validate((SEntity)target, docItem, typemap);

            // entity rule is inapplicable if any attribute rules are inapplicable
            if (result == null)
                return null;

            // entity rule fails if any attribute rules fail
            if (!result.Value)
                return false;
        }
    }

    return true;
}

```

4.7.7.3.3 Testing tools sample files

Sample IFC files are available at the following URL: http://www.nibs.org/?page=bsa_commonbimfiles

4.7.7.3.4 Testing tool software availability

The IFCDOC tool for testing is available at the following URL: <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-beta-summary9>

4.7.8 Implementation resources

4.7.8.1 Implementation resources list

4.7.8.1.1 Implementation guides

Implementation guides are available at buildingsmart-tech.org/downloads.

4.7.8.2 Implementation resources completeness

4.7.8.2.1 Workflow coverage methodology

Implementation resources for various BIM platform workflows are available at <http://www.buildingsmart-tech.org/implementation>. Additional platforms and workflows may be added as indicated on the web page.

4.7.8.2.2 Workflow coverage analysis

A map of software applications, platforms, and supported model views is available at <http://www.buildingsmart-tech.org/implementation/implementations>.

4.7.9 Revision Plans

4.7.9.1 Revision plans list

4.7.9.1.1 Revision management process

No revisions to this version 1.0 standard have been identified at this time. Revisions will be identified, evaluated and introduced based on initial uses of LCie exchanges that will begin as part of 2015 Alliance Challenge events.

4.7.9.1.2 Revision management notification

Revisions will be proposed through a new LinkedIn LCie Group.

4.7.9.2 Proposed revision deployment methods

4.7.9.2.1 Revision management process

No revisions to this version 1.0 standard have been identified at this time. Revisions will be identified, evaluated and introduced based on initial uses of LCie exchanges that will begin as part of 2015 Alliance Challenge events.

4.7.9.2.2 Revision management notification

Revisions will be proposed through a new LinkedIn LCie Group.

[Annex A](#)

A.1 The following electronic formats are attached herein:

SPARKie2013.exp – schema in EXPRESS format

SPARKie2013.xsd – schema in XSD format

SPARKie2013.mdxml – model view definition

SPARKie2013.ifc – property set templates

Bibliography

ISO 639-1, Codes for the representation of names of languages — Part 1: Alpha-2 code

ISO 639-2, Codes for the representation of names of languages — Part 2: Alpha-3 code

ISO 639-3, Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages

ISO 6707-1, Building and civil engineering — Vocabulary — Part 1: General terms

ISO 8601, Data elements and interchange formats — Information Exchange — Representation of dates and times.

ISO 10303-1:1994, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles

ISO 10303-11, Industrial automation systems and integration — Product data representation and exchange — Part 11: description methods: The EXPRESS Language Reference Manual

ISO 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure

ISO 10303-28, Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas

ISO 10303-41, Product data representation and exchange — Integrated generic resource — Fundamentals of product description and support

ISO 10303-42, Product data representation and exchange — Integrated generic resource — Geometric and topological representation

ISO 10303-43, Product data representation and exchange — Integrated generic resource — Representation structures

ISO 10303-46, Product data representation and exchange — Integrated generic resource — Visual presentation

ISO 10303-514, Product data representation and exchange — Application interpreted construct — Advanced boundary representation

ISO 12006-3, Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information

ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation.

ISO/IEC 14772-1, Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language — Part 1: Functional specification and UTF-8 encoding

ISO/IEC 19775-1, Information technology — Computer graphics and image processing — Extensible 3D (X3D) — Part 1: Architecture and base components

ISO/IEC 81346-12, Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 12: Buildings and building services

CSS-1, Cascading Style Sheets, level 1 — W3C Recommendation

XML Schema Part 2, XML Schema Part 2: Datatypes — W3C Recommendation

RFC 3986, Uniform Resource Identifier (URI): Generic Syntax — Network Working Group NWG Standard

RFC 5646, Tags for Identifying Languages — Internet Engineering Task Force IETF Best Current Practice 47