

Chapter 4

Infra IFC Entities Including IfcRoad Their Types List V1.0

July. 01 2016

Korea Institute of Civil Engineering
and Building Technology

Full Entity Lists of Infra IFC including IfcRoad

| | Entity | | | | |
|----------------------------------|-----------|---------------------------------|--|--|--|
| | Eng. Name | Entity List | | | |
| Civil Element | | IfcCivilElement | | | |
| Road Element | | IfcRoadElement_K | | | |
| Road Body | | IfcRoadBody_K | | | |
| Shoulder | | IfcRoadShoulder_K | | | |
| Median Strip | | IfcRoadMedianStrip_K | | | |
| Curb | | IfcCurb_K | | | |
| Pavement | | IfcRoadPavement_K | | | |
| Bridge Element | | IfcBridgeElement_K | | | |
| Deck | | IfcBridgeDeck_K | | | |
| Span | | IfcBridgeSpan_K | | | |
| Segment | | IfcBridgeSegment_K | | | |
| Tower | | IfcBridgeTower_K | | | |
| Cable | | IfcBridgeCable_K | | | |
| Pier | | IfcBridgePier_K | | | |
| Abutment | | IfcBridgeAbutment_K | | | |
| Coping | | IfcBridgeCoping_K | | | |
| Girder | | IfcBridgeGirder_K | | | |
| Tunnel Element | | IfcTunnelElement_K | | | |
| Lining | | IfcTunnelLining_K | | | |
| Lining Segment | | IfcTunnelLiningSegment_K | | | |
| Civil Structure Element | | IfcCivilStructureElement_K | | | |
| Culvert | | IfcCulvert_K | | | |
| Retaining Wall | | IfcRetainingWall_K | | | |
| Caisson | | IfcCaisson_K | | | |
| Subsidiary Facility | | IfcSubsidiaryFacility_K | | | |
| Road Sign Element | | IfcRoadSignElement_K | | | |
| Road Guard | | IfcGuard_K | | | |
| Road Pavement Addition | | IfcPavementAddition_K | | | |
| Earthwork Element | | IfcEarthworkElement_K | | | |
| Civil Element Proxy | | IfcCivilElementProxy_K | | | |
| Spatial Element | | IfcSpatialElement | | | |
| Spatial Zone | | IfcSpatialZone | | | |
| Civil Structure Element | | IfcCivilStructureElement_K | | | |
| Road | | IfcRoad_K | | | |
| Bridge | | IfcBridge_K | | | |
| Tunnel | | IfcTunnel_K | | | |
| Civil Spatial Boundary | | IfcCivilSpatialBoundary_K | | | |
| Node Space | | IfcCurvilinearNodeSpace_K | | | |
| Vertical Subspace | | IfcVerticalSubspace_K | | | |
| Linear Reference Space | | IfcLinearRefSpace_K | | | |
| Distribution Flow Element | | IfcDistributionFlowElement | | | |
| Flow Segment | | IfcFlowSegment | | | |
| Pipe Segment | | IfcPipeSegment | | | |
| Gutter Segment | | IfcGutterSegment_K | | | |
| Flow Fitting | | IfcFlowFitting | | | |
| Gutter Fitting | | IfcGutterFitting_K | | | |
| Component Element | | IfcElementComponent | | | |
| Discrete Accessory | | IfcDiscreteAccessory | | | |
| Reinforcing Element | | IfcReinforcingElement | | | |
| Ground Reinforcing Element | | IfcGroundReinforcingElement_K | | | |
| Water Proofing Element | | IfcWaterProofingElement_K | | | |
| Road Element Part | | IfcRoadElementPart_K | | | |
| Bridge Element Part | | IfcBridgeElementPart_K | | | |
| Tunnel Element Part | | IfcTunnelElementPart_K | | | |
| Construction Resource | | IfcConstructionResource | | | |
| Construction Material Resource | | IfcConstructionMaterialResource | | | |
| Earthwork Material Resource | | IfcEarthworkMaterialResource_K | | | |
| Building Element | | IfcBuildingElement | | | |
| Slab | | IfcSlab | | | |
| Wall | | IfcWall | | | |
| Footing | | IfcFooting | | | |

Type Entity List with Type Enumeration

| | Entity | | Data Type | | Type Enumeration |
|---------------------------|--------|------------------------------|-----------------------------------|--|--------------------------|
| | Name | Type Entity | English Name | | |
| Civil Element Type | | IfcCivilElementType | | | |
| Road Element Type | | IfcRoadElementType_K | | | |
| Road Body Type | | IfcRoadBodyType_K | | | |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Shoulder Type | | IfcRoadShoulderType_K | | | |
| | | | Full Width | | FULLWIDTH_K |
| | | | Half Width | | HALFWIDTH_K |
| | | | Narrow Width | | NARROWWIDTH_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Median Strip Type | | IfcRoadMedianStripType_K | | | |
| | | | Concrete Guard Fence | | GUARDFENCEOFCONCRETE_K |
| | | | Guardrail | | GUARDRAIL_K |
| | | | Green Area | | GREENAREA_K |
| | | | Concrete Curb | | CURBOFCONCRETE_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Curb Type | | IfcCurbType_K | | | |
| | | | Mountable Curb(Gentle Slope Type) | | MOUNTABLE_CURB_K |
| | | | Barrier1 Curb(Steep Slope Type) | | BARRIER1_K |
| | | | Barrier2(Rectangle Type) | | BARRIER2_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Pavement Type | | IfcRoadPavementType_K | | | |
| | | | Surface Layer | | SURFACE_K |
| | | | Intermediate Course | | INTERMEDIATECOURSE_K |
| | | | Subbase Course | | SUBBASE_K |
| | | | Base Course | | BASECOURSE_K |
| | | | Lean Concrete | | LEANCONCRETE_K |
| | | | Anti Freezing Layer | | ANTIFREEZINGLAYER_K |
| | | | Bridge Deck Surfacing | | BRIDGEDECKSURFACING_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Bridge Element Type | | IfcBridgeElementType_K | | | |
| Deck Type | | IfcBridgeDeckType_K | | | |
| | | | Approach Slab | | APPROACHSLAB_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Span Type | | IfcBridgeSpanType_K | | | |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Segment Type | | IfcBridgeSegmentType_K | | | |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Tower Type | | IfcBridgeTowerType_K | | | |
| | | | A Shaped Type | | A_SHAPED_K |
| | | | H Shaped Type | | H_SHAPED_K |
| | | | I Shaped Type | | I_SHAPED_K |
| | | | Diamond Shaped type | | DIAMOND_SHAPED_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Cable Type | | IfcBridgeCableType_K | | | |
| | | | Suspender | | SUSPENDER_K |
| | | | Suspension Cable | | SUSPENSIONCABLE_K |
| | | | Tension Cable | | TENSIONCABLE_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Pier Type | | IfcBridgePierType_K | | | |
| | | | Gravity Type | | GRAVITY_TYPE_K |
| | | | Wall Type | | WALL_TYPE_K |
| | | | Rahmen Pier Type | | RAHMEN_PIER_TYPE_K |
| | | | T Shaped Type | | T_SHAPED_TYPE_K |
| | | | Rahmen Abut Type | | RAHMEN_ABUT_TYPE_K |
| | | | Arch Type | | ARCH_TYPE_K |
| | | | V Shaped Type | | V_SHAPED_TYPE_K |
| | | | Semi Gravity Type | | SEMI_GRAVITY_TYPE_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Abutment Type | | IfcBridgeAbutmentType_K | | | |
| | | | Gravity Type | | GRAVITY_TYPE_K |
| | | | Semi Gravity Type | | SEMI_GRAVITY_TYPE_K |
| | | | Reversed T Shaped Type | | REVERSED_T_SHAPED_TYPE_K |
| | | | Counterfort | | COUNTERFORT_TYPE_K |
| | | | Rahmen Type | | RAHMEN_TYPE_K |
| | | | Rahmen Abut Type | | RAHMEN_ABUT_TYPE_K |
| | | | Box Type | | BOX_TYPE_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Coping Type | | IfcBridgeCopingType_K | | | |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Girder Type | | IfcBridgeGirderType_K | | | |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Tunnel Element Type | | IfcTunnelElementType_K | | | |
| Lining Type | | IfcTunnelLiningType_K | | | |
| | | | Ground Reinforcing Body Type | | GROUNDREINFORCING_BODY_K |
| | | | Shotcrete Body Type | | SHOTCRETE_BODY_K |
| | | | Lining Body Type | | LINING_BODY_K |
| | | | User Defined Type | | USERDEFINED |
| | | | Not Defined Type | | NOTDEFINED |
| Lining Segment Type | | IfcTunnelLiningSegmentType_K | | | |
| | | | Steel Segment Type | | STEEL_SEGMENT_K |
| | | | Concrete Segment Type | | CONCRETE_SEGMENT_K |
| | | | Cast Iron Segment Type | | CAST_IRON_SEGMENT_K |
| | | | Combined Segment Type | | COMBINED_SEGMENT_K |

| | | | | | |
|------------------------------|--|--|--|---------------------------------|-------------------------------|
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Civil Structure Element Type | | | | IfcCivilStructureElementType_K | |
| Culvert Type | | | | IfcCulvertType_K | |
| | | | | One-way Waterway Type | ONEWAY_WATERWAY_K |
| | | | | Two-way Waterway Type | TWOWAY_WATERWAY_K |
| | | | | Three-way Waterway Type | THREEWAY_WATERWAY_K |
| | | | | One-way Passageway Type | ONEWAY_PASSAGEWAY_K |
| | | | | Two-way Passageway Type | TWOWAY_PASSAGEWAY_K |
| | | | | One-way Skewed Waterway Type | ONEWAY_SKEWED_WATERWAY_K |
| | | | | Two-way Skewed Waterway Type | TWOWAY_SKEWED_WATERWAY_K |
| | | | | Three-way Skewed Waterway Type | THREEWAY_SKEWED_WATERWAY_K |
| | | | | One-way Skewed Passageway Type | ONEWAY_SKEWED_PASSAGEWAY_K |
| | | | | Two-way Skewed Passageway Type | TWOWAY_SKEWED_PASSAGEWAY_K |
| | | | | Stone filled Culvert Type1 | STONEFILLED_CULVERT_TYPE1_K |
| | | | | Stone filled Culvert Type2 | STONEFILLED_CULVERT_TYPE2_K |
| | | | | Stone filled Culvert Type3 | STONEFILLED_CULVERT_TYPE3_K |
| | | | | Stone filled Culvert Type4 | STONEFILLED_CULVERT_TYPE4_K |
| | | | | Stone filled Culvert Type5 | STONEFILLED_CULVERT_TYPES_K |
| | | | | Common Duct Type | COMMONDUCT_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Retaining Wall Type | | | | IfcRetainingWallType_K | |
| | | | | Gravity Type | GRAVITY_TYPE_K |
| | | | | Semi Gravity Type | SEMIGRAVITY_TYPE_K |
| | | | | Non Standing Type | NON_STANDING_K |
| | | | | Cantilever Type | CANTILEVER_TYPE_K |
| | | | | Reversed T Shaped Type | REVERSED_T_SHAPED_K |
| | | | | L Shaped Type | L_SHAPED_K |
| | | | | Reversed L Shaped Type | REVERSED_L_SHAPED_K |
| | | | | Counterfort type | COUNTERFORT_K |
| | | | | Masonry Wall type | MASONARY_K |
| | | | | Reinforced Earth Retaining Wall | REINFORCED_EARTH_K |
| | | | | Block Type | BLOCK_TYPE_K |
| | | | | Panel type | PANEL_TYPE_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Caisson Type | | | | IfcCaissonType_K | |
| | | | | Open Caisson Type | OPEN_K |
| | | | | Box Caisson Type | BOX_K |
| | | | | Pneumatic Caisson Type | PNEUMATIC_K |
| | | | | Floating Caisson Type | FLOATING_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Subsidiary Facility Type | | | | IfcSubsidiaryFacilityType_K | |
| Road Sign Element Type | | | | IfcRoadSignElementType_K | |
| | | | | Delineator | DELINEATOR_K |
| | | | | Traffic Sign | TRAFFICSIGN_K |
| | | | | Reflecting Mirror | REFLECTINGMIRROR_K |
| | | | | Traffic Signal | TRAFFICSIGNAL_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Road Guard Type | | | | IfcGuardType_K | |
| | | | | Crash Cushion | CRASHCUSHION_K |
| | | | | Sound Proof Wall | SOUNDPROOF_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Road Pavement Addition Type | | | | IfcPavementAdditionType_K | |
| | | | | Pedestrian Crosswalk | PEDESTRIANCROSSWALK_K |
| | | | | Road Marker | ROADMARKER_K |
| | | | | Speed Hump | SPEEDHUMP_K |
| | | | | Anti Sliding | ANTISLIDING_K |
| | | | | Rumble Strip | RUMBLESTRIP_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Earthwork Element Type | | | | IfcEarthworkElementType_K | |
| | | | | Cutting | CUTTING_K |
| | | | | Filling | FILLING_K |
| | | | | Cutting and Filling | CUTTINGANDFILLING_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Civil Element Proxy Type | | | | IfcCivilElementProxyType_K | |
| Spatial Element Type | | | | IfcSpatialElementType | |
| Spatial Zone Type | | | | IfcSpatialZoneType | |
| | | | | Civil Structure | CIVIL_STRUCTURE_K |
| Civil Structure Element Type | | | | IfcCivilStructureElementType_K | |
| Road Type | | | | IfcRoadType_K | |
| | | | | Ramp | RAMPROAD_K |
| | | | | Approach Road | APPROACHROAD_K |
| | | | | User Defined Type | USERDEFINED |
| | | | | Not Defined Type | NOTDEFINED |
| Bridge Type | | | | IfcBridgeType_K | |
| | | | | RC Slab Bridge | RC_SLAB_BRIDGE_K |
| | | | | RC Hollow Core Slab Bridge | RC_HOLLOW_CORE_SLAB_BRIDGE_K |
| | | | | RC T Beam Girder Bridge | RC_T_BEAM_GIRDER_BRIDGE_K |
| | | | | Steel I Beam Girder Bridge | STEEL_I_BEAM_GIRDER_BRIDGE_K |
| | | | | PSC Box Girder Bridge | PSC_BOX_GIRDER_BRIDGE_K |
| | | | | Steel Plate Girder Bridge | STEEL_PLATE_GIRDER_BRIDGE_K |
| | | | | Rahmen Bridge | RAHMEN_BRIDGE_K |
| | | | | Truss Bridge | TRUSS_BRIDGE_K |
| | | | | Arch Bridge | ARCH_BRIDGE_K |
| | | | | Cable Stayed Bridge | CABLE_STAYED_BRIDGE_K |
| | | | | Suspension Bridge | SUSPENSION_BRIDGE_K |
| | | | | Preflex Girder Bridge | PREFLEX_GIRDER_BRIDGE_K |
| | | | | PSC I Girder Bridge | PSC_I_GIRDER_BRIDGE_K |
| | | | | PSC Slab Bridge | PSC_SLAB_BRIDGE_K |
| | | | | PSC Hollow Core Slab Bridge | PSC_HOLLOW_CORE_SLAB_BRIDGE_K |
| | | | | RC Box Girder Bridge | RC_BOX_GIRDER_BRIDGE_K |
| | | | | Steel Box Girder Bridge | STEEL_BOX_GIRDER_BRIDGE_K |
| | | | | Overpass Bridge | OVERPASS_BRIDGE_K |

| | | | | | | |
|---------------------------------------|--|--|--|--|-----------------------------------|------------------------------|
| | | | | | Ramp Bride | RAMP_BRIDGE_K |
| | | | | | Approach Bridge | APPROACH_BRIDGE_K |
| | | | | | Extra dosed Bridge | EXTRADOSED_BRIDGE_K |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Tunnel Type | | | | | IfcTunnelType | |
| | | | | | NATM Type | NATM_K |
| | | | | | TBM Shield Type | TBM_SHIELD_K |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Civil Spatial Boundary Type | | | | | IfcCivilSpatialBoundaryType_K | |
| Node Space Type | | | | | IfcCurvilinearNodeSpaceType_K | |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Vertical Subspace Type | | | | | IfcVerticalSubspaceType_K | |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Linear Reference Space Type | | | | | IfcLinearRefSpaceType_K | |
| | | | | | Frontage Road | FRONTAGEROAD_K |
| | | | | | Footpath | ROADFOOTPATH_K |
| | | | | | Marginal Strip | ROADMARGINALSTRIP |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Distribution Flow Element Type | | | | | IfcDistributionFlowElementType | |
| Flow Segment Type | | | | | IfcFlowSegmentType | |
| Pipe Segment Type | | | | | IfcPipeSegmentType | |
| | | | | | Closed Conduit | CLOSEDCONDUIT_K |
| | | | | | Perforated | PERFORATED_K |
| | | | | | Hume Segment | HUMESEMENT_K |
| Gutter Segment Type | | | | | IfcGutterSegmentType_K | |
| | | | | | V Shaped Type | V_TYPE_K |
| | | | | | L Shaped Type | L_TYPE_K |
| | | | | | U Shaped Type | U_TYPE_K |
| | | | | | Canal Type | CANAL_K |
| | | | | | Open Channel Type | OPENCHANNEL_K |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Flow Fitting Type | | | | | IfcFlowFittingType | |
| Gutter Fitting Type | | | | | IfcGutterFittingType_K | |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| | | | | | IfcDistributionChamberElementType | |
| | | | | | Collecting Well | COLLECTINGWELL_K |
| | | | | | Catch Drain | CATCHDRAIN_K |
| Component Element Type | | | | | IfcElementComponentType | |
| Discrete Accessory Type | | | | | IfcDiscreteAccessoryType | |
| | | | | | Steel Grating | STEELGRATING_K |
| Reinforcing Element Type | | | | | IfcReinforcingElementType | |
| 지반보강요소유형 | | | | | IfcGroundReinforcingElementType_K | |
| | | | | | Earth Anchor | EARTH_ANCHOR_K |
| | | | | | Rock Bolt | ROCK_BOLT_K |
| | | | | | Rock Anchor | ROCK_ANCHOR_K |
| | | | | | Shotcrete | SHOTCRETE_K |
| | | | | | Steeel Rib | STEEL_RIB_K |
| | | | | | Steel Fiber | STEEL_FIBER_K |
| | | | | | Synthetic Fiber | SYNTHETIC_FIBER_K |
| | | | | | Pile | PILE_K |
| | | | | | Wire Rope | WIREROPE_K |
| | | | | | Block | BLOCK_K |
| | | | | | Concrete | CONCRETE_K |
| | | | | | Stone | STONE_K |
| | | | | | Textile | TEXTILE_K |
| | | | | | Net | NET_K |
| | | | | | Plant | PLANT_K |
| | | | | | Counter Weight Fill | COUNTERWEIGHT_FILL_K |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Water Proofing Element Type | | | | | IfcWaterProofingElementType_K | |
| | | | | | Water Proof Sheet | WATERPROOF_SHEET_K |
| | | | | | Fluid Applied Water Proof | FLUID_APPLIED_K |
| | | | | | Asphalt Water Proof | ASPHALT_K |
| | | | | | Sealing | SEALING_K |
| | | | | | Caulking | CAULKING_K |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Road Element Part Type | | | | | IfcRoadElementPartType_K | |
| | | | | | Stabilization Filter | STABILIZATIONFILTER_K |
| | | | | | Spacer | SPACER_K |
| | | | | | Rail Joint | RAILJOINT_K |
| | | | | | Expansion Joint | EXPANSIONJOINT_K |
| | | | | | Transverse Contraction Joint | TRANSVERSECONTRACTIONJOINT_K |
| | | | | | Longitudinal Joint | LONGITUDINALJOINT_K |
| | | | | | Construction Joint | CONSTRUCTIONJOINT_K |
| | | | | | Styrofoam | STYROFOAM_K |
| | | | | | Wire mesh | WIREMESH_K |
| | | | | | Prime Coat | PRIMECOAT_K |
| | | | | | Tack Coat | TACKCOAT_K |
| | | | | | Geo Textile | GEOTEXTILE_K |
| | | | | | Dower Bar | DOWERBAR_K |
| | | | | | Tie Bar | TIEBAR_K |
| | | | | | Slip Bar | SLIPBAR_K |
| | | | | | Bed Joint | BEDJOINT_K |
| | | | | | Cutting Joint | CUTTINGJOINT_K |
| | | | | | Foot Path Boundary Stone | FOOTPATHBOUNDARYSTONE_K |
| | | | | | Road Boundary Stone | ROADBOUNDARYSTONE_K |
| | | | | | Inspection Ladder | INSPECTIONLADDER_K |
| | | | | | User Defined Type | USERDEFINED |
| | | | | | Not Defined Type | NOTDEFINED |
| Bridge Element Part Type | | | | | IfcBridgeElementPartType_K | |
| | | | | | Flange | FLANGE_K |
| | | | | | Overhang | OVERHANG_K |

| | | | | | | | |
|--|-------------------------------------|--|--|--|--|-------------------------------------|-------------------------------|
| | | | | | | Flooring | FLOORING_K |
| | | | | | | Vertical Stiffener | VERTICAL_STIFFENER_K |
| | | | | | | Longitudinal Stiffener | LONGITUDINAL_STIFFENER_K |
| | | | | | | Web | WEB_K |
| | | | | | | Rib | RIB_K |
| | | | | | | Steel Plate | PLATE_K |
| | | | | | | Strand | STRAND_K |
| | | | | | | Bracing | BRACING_K |
| | | | | | | Girder Linkage | GIRDERLINKAGE_K |
| | | | | | | Anchorage | ANCHORAGE_K |
| | | | | | | Pier Scour Protection | PIER_SCOUR_PROTECTION_K |
| | | | | | | Ship Impact Protection | SHIP_IMPACT_PROTECTION_K |
| | | | | | | User Defined Type | USERDEFINED |
| | | | | | | Not Defined Type | NOTDEFINED |
| | Tunnel Element Part Type | | | | | IfcTunnelElementPartType_K | |
| | | | | | | Segment Joint | SEGMENT_JOINT_K |
| | | | | | | Ring Joint | RING_JOINT_K |
| | | | | | | User Defined Type | USER_DEFINED |
| | | | | | | Not Defined Type | NOTDEFINED |
| | Construction Resource Type | | | | | IfcConstructionResourceType | |
| | Construction Material Resource Type | | | | | IfcConstructionMaterialResourceType | |
| | | | | | | Asphalt | ASPHALT_K |
| | | | | | | Granite | GRANITE_K |
| | | | | | | Plywood | PLYWOOD_K |
| | | | | | | Rubble | RUBBLE_K |
| | | | | | | Ascon | ASCON_K |
| | | | | | | Broken Stone | BROKENSTONE_K |
| | | | | | | Stopper Stone | STOPPERSTONE_K |
| | | | | | | LMC | LATEXMODIFIEDCONCRETE_K |
| | Earthwork Material Resource Type | | | | | IfcEarthworkMaterialResourceType_K | |
| | | | | | | Gravel | GRAVEL_K |
| | | | | | | Sand | SAND_K |
| | | | | | | Silt | SILT_K |
| | | | | | | Clay | CLAY_K |
| | | | | | | Organic Clay | ORGANIC_CLAY_K |
| | | | | | | Peat | PEAT_K |
| | | | | | | Reaping Rock | REAPINGROCK_K |
| | | | | | | Blasting Rock-Soft Rock | BLASTINGROCK_SOFT_K |
| | | | | | | Blasting Rock-Normal Rock | BLASTINGROCK_NORMAL_K |
| | | | | | | Blasting Rock-Hard Rock | BLASTINGROCK_HARD_K |
| | | | | | | User Defined Type | USERDEFINED |
| | | | | | | Not Defined Type | NOTDEFINED |
| | Building Element Type | | | | | IfcBuildingElementType | |
| | Slab Type | | | | | IfcSlabType | |
| | | | | | | Invert | INVERT_K |
| | Wall Type | | | | | IfcWallType | |
| | | | | | | Wing Wall | WINGWALL_K |
| | | | | | | Chest Wall | CHESTWALL_K |
| | Footing Type | | | | | IfcFootingType | |
| | | | | | | Base Footing | BASE_FOOTING_K |
| | | | | | | Pit mouth Footing | PITMOUTH_FOOTING_K |
| | | | | | | PSC Pile Foundation | PSC_PILE_FOUNDATION_K |
| | | | | | | RC Pile Foundation | RC_PILE_FOUNDATION_K |
| | | | | | | Slurry Wall Foundation | SLURRY_WALL_FOUNDATION_K |
| | | | | | | Spread Foundation | SPREAD_FOUNDATION_K |
| | | | | | | Steel Pile Foundation | STEEL_PILE_FOUNDATION_K |
| | | | | | | Steel Sheet Pile Foundation | STEEL_SHEET_PILE_FOUNDATION_K |
| | | | | | | Mat Foundation | MAT_FOUNDATION_K |

Added into new types for civil facilities in existing building element type

Chapter 5

Infra IFC Schema based on IFC4 ADD1 with IfcAlignment by KICT (Express) V1.0

July. 01 2016

**Korea Institute of Civil Engineering
and Building Technology**

SCHEMA Ifc4;

TYPE IfcStrippedOptional = BOOLEAN;

END_TYPE;

TYPE IfcAbsorbedDoseMeasure = REAL;

END_TYPE;

TYPE IfcAccelerationMeasure = REAL;

END_TYPE;

TYPE IfcAmountOfSubstanceMeasure = REAL;

END_TYPE;

TYPE IfcAngularVelocityMeasure = REAL;

END_TYPE;

TYPE IfcArcIndex = LIST [3:3] OF IfcPositiveInteger;

END_TYPE;

TYPE IfcPositiveInteger = IfcInteger;

WHERE

WR1 : SELF > 0;

END_TYPE;

TYPE IfcInteger = INTEGER;

END_TYPE;

TYPE IfcAreaDensityMeasure = REAL;

END_TYPE;


```
TYPE IfcAreaMeasure = REAL;
```

```
END_TYPE;
```

```
TYPE IfcBinary = BINARY;
```

```
END_TYPE;
```

```
TYPE IfcBoolean = BOOLEAN;
```

```
END_TYPE;
```

```
TYPE IfcBoxAlignment = IfcLabel;
```

```
WHERE
```

```
WR1 : SELF IN ['top-left', 'top-middle', 'top-right', 'middle-left', 'center', 'middle-right', 'bottom-left', 'bottom-middle', 'bottom-right'];
```

```
END_TYPE;
```

```
TYPE IfcLabel = STRING (255);
```

```
END_TYPE;
```

```
TYPE IfcCardinalPointReference = INTEGER;
```

```
WHERE
```

```
GREATERTHANZERO : SELF > 0;
```

```
END_TYPE;
```

```
TYPE IfcComplexNumber = ARRAY [1:2] OF REAL;
```

```
END_TYPE;
```

```
TYPE IfcCompoundPlaneAngleMeasure = LIST [3:4] OF INTEGER;
```

```
WHERE
```

```
MINUTESINRANGE : ABS(SELF[2]) < 60;
```

```

SECOND SINRANGE      : ABS(SELF[3]) < 60;
MICROSECONDSINRANGE : (SIZEOF(SELF) = 3) OR (ABS(SELF[4]) < 1000000);
CONSISTENTSIGN      : ((SELF[1] >= 0) AND (SELF[2] >= 0) AND (SELF[3] >= 0) AND ((SIZEOF(SELF)
= 3) OR (SELF[4] >= 0)))
OR
((SELF[1] <= 0) AND (SELF[2] <= 0) AND (SELF[3] <= 0) AND ((SIZEOF(SELF)
= 3) OR (SELF[4] <= 0)));
END_TYPE;

```

```

TYPE IfcContextDependentMeasure = REAL;
END_TYPE;

```

```

TYPE IfcCountMeasure = NUMBER;
END_TYPE;

```

```

TYPE IfcCurvatureMeasure = REAL;
END_TYPE;

```

```

TYPE IfcDate = STRING;
END_TYPE;

```

```

TYPE IfcDateTime = STRING;
END_TYPE;

```

```

TYPE IfcDayInMonthNumber = INTEGER;
WHERE
  VALIDRANGE : {1 <= SELF <= 31};
END_TYPE;

```

```

TYPE IfcDayInWeekNumber = INTEGER;

```

```
WHERE
  VALDRANGE : {1 <= SELF <= 7};
END_TYPE;

TYPE IfcDescriptiveMeasure = STRING;
END_TYPE;

TYPE IfcDimensionCount = INTEGER;
  WHERE
    WR1 : { 0 < SELF <= 3 };
END_TYPE;

TYPE IfcDoseEquivalentMeasure = REAL;
END_TYPE;

TYPE IfcDuration = STRING;
END_TYPE;

TYPE IfcDynamicViscosityMeasure = REAL;
END_TYPE;

TYPE IfcElectricCapacitanceMeasure = REAL;
END_TYPE;

TYPE IfcElectricChargeMeasure = REAL;
END_TYPE;

TYPE IfcElectricConductanceMeasure = REAL;
END_TYPE;
```

```
TYPE IfcElectricCurrentMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcElectricResistanceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcElectricVoltageMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcEnergyMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcFontStyle = STRING;  
  WHERE  
    WR1 : SELF IN ['normal','italic','oblique'];  
END_TYPE;
```

```
TYPE IfcFontVariant = STRING;  
  WHERE  
    WR1 : SELF IN ['normal','small-caps'];  
END_TYPE;
```

```
TYPE IfcFontWeight = STRING;  
  WHERE  
    WR1 : SELF IN ['normal','small-caps','100','200','300','400','500','600','700','800','900'];  
END_TYPE;
```

```
TYPE IfcForceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcFrequencyMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcGloballyUniqueId = STRING (22) FIXED;  
END_TYPE;
```

```
TYPE IfcHeatFluxDensityMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcHeatingValueMeasure = REAL;  
  WHERE  
    WR1 : SELF > 0.;  
END_TYPE;
```

```
TYPE IfcIdentifier = STRING (255);  
END_TYPE;
```

```
TYPE IfcIlluminanceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcInductanceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcIntegerCountRateMeasure = INTEGER;  
END_TYPE;
```

```
TYPE IfcIonConcentrationMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcIsothermalMoistureCapacityMeasure = REAL;
```

END_TYPE;

TYPE IfcKinematicViscosityMeasure = REAL;

END_TYPE;

TYPE IfcLanguageId = IfcIdentifier;

END_TYPE;

TYPE IfcLengthMeasure = REAL;

END_TYPE;

TYPE IfcLineIndex = LIST [2:?] OF IfcPositiveInteger;

END_TYPE;

TYPE IfcLinearForceMeasure = REAL;

END_TYPE;

TYPE IfcLinearMomentMeasure = REAL;

END_TYPE;

TYPE IfcLinearStiffnessMeasure = REAL;

END_TYPE;

TYPE IfcLinearVelocityMeasure = REAL;

END_TYPE;

TYPE IfcLogical = LOGICAL;

END_TYPE;

TYPE IfcLuminousFluxMeasure = REAL;

END_TYPE;

TYPE IfcLuminousIntensityDistributionMeasure = REAL;

END_TYPE;

TYPE IfcLuminousIntensityMeasure = REAL;

END_TYPE;

TYPE IfcMagneticFluxDensityMeasure = REAL;

END_TYPE;

TYPE IfcMagneticFluxMeasure = REAL;

END_TYPE;

TYPE IfcMassDensityMeasure = REAL;

END_TYPE;

TYPE IfcMassFlowRateMeasure = REAL;

END_TYPE;

TYPE IfcMassMeasure = REAL;

END_TYPE;

TYPE IfcMassPerLengthMeasure = REAL;

END_TYPE;

TYPE IfcModulusOfElasticityMeasure = REAL;

END_TYPE;

TYPE IfcModulusOfLinearSubgradeReactionMeasure = REAL;

END_TYPE;

TYPE IfcModulusOfRotationalSubgradeReactionMeasure = REAL;

END_TYPE;

TYPE IfcModulusOfSubgradeReactionMeasure = REAL;

END_TYPE;

TYPE IfcMoistureDiffusivityMeasure = REAL;

END_TYPE;

TYPE IfcMolecularWeightMeasure = REAL;

END_TYPE;

TYPE IfcMomentOfInertiaMeasure = REAL;

END_TYPE;

TYPE IfcMonetaryMeasure = REAL;

END_TYPE;

TYPE IfcMonthInYearNumber = INTEGER;

WHERE

VALIDRANGE : {1 <= SELF <= 12};

END_TYPE;

TYPE IfcNonNegativeLengthMeasure = IfcLengthMeasure;

WHERE

NOTNEGATIVE : SELF >= 0.;

END_TYPE;


```
TYPE IfcNormalisedRatioMeasure = IfcRatioMeasure;  
  WHERE  
    WR1 : {0.0 <= SELF <= 1.0};  
END_TYPE;
```

```
TYPE IfcRatioMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcNumericMeasure = NUMBER;  
END_TYPE;
```

```
TYPE IfcPHMeasure = REAL;  
  WHERE  
    WR21 : {0.0 <= SELF <= 14.0};  
END_TYPE;
```

```
TYPE IfcParameterValue = REAL;  
END_TYPE;
```

```
TYPE IfcPlanarForceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcPlaneAngleMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcPositiveLengthMeasure = IfcLengthMeasure;  
  WHERE  
    WR1 : SELF > 0.;  
END_TYPE;
```

```
TYPE IfcPositivePlaneAngleMeasure = IfcPlaneAngleMeasure;
```

```
WHERE
```

```
WR1 : SELF > 0.;
```

```
END_TYPE;
```

```
TYPE IfcPositiveRatioMeasure = IfcRatioMeasure;
```

```
WHERE
```

```
WR1 : SELF > 0.;
```

```
END_TYPE;
```

```
TYPE IfcPowerMeasure = REAL;
```

```
END_TYPE;
```

```
TYPE IfcPresentableText = STRING;
```

```
END_TYPE;
```

```
TYPE IfcPressureMeasure = REAL;
```

```
END_TYPE;
```

```
TYPE IfcPropertySetDefinitionSet = SET [1:?] OF IfcPropertySetDefinition;
```

```
END_TYPE;
```

```
TYPE IfcGeometricProjectionEnum = ENUMERATION OF
```

```
(GRAPH_VIEW,
```

```
SKETCH_VIEW,
```

```
MODEL_VIEW,
```

```
PLAN_VIEW,
```

```
REFLECTED_PLAN_VIEW,
```

```
SECTION_VIEW,
```

```
ELEVATION_VIEW,
```

```
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAxis2Placement = SELECT  
  (IfcAxis2Placement2D,  
   IfcAxis2Placement3D);  
END_TYPE;
```

```
TYPE IfcText = STRING;  
END_TYPE;
```

```
TYPE IfcProductRepresentationSelect = SELECT  
  (IfcProductDefinitionShape,  
   IfcRepresentationMap);  
END_TYPE;
```

```
TYPE IfcLayerSetDirectionEnum = ENUMERATION OF  
  (AXIS1,  
   AXIS2,  
   AXIS3);  
END_TYPE;
```

```
TYPE IfcReal = REAL;  
END_TYPE;
```

```
TYPE IfcKnotType = ENUMERATION OF  
  (UNIFORM_KNOTS,  
   QUASI_UNIFORM_KNOTS,  
   PIECEWISE_BEZIER_KNOTS,
```

UNSPECIFIED);
END_TYPE;

TYPE IfcBSplineSurfaceForm = ENUMERATION OF
(PLANE_SURF,
CYLINDRICAL_SURF,
CONICAL_SURF,
SPHERICAL_SURF,
TOROIDAL_SURF,
SURF_OF_REVOLUTION,
RULED_SURF,
GENERALISED_CONE,
QUADRATIC_SURF,
SURF_OF_LINEAR_EXTRUSION,
UNSPECIFIED);
END_TYPE;

TYPE IfcTransitionCode = ENUMERATION OF
(DISCONTINUOUS,
CONTINUOUS,
CONTSAMEGRADIENT,
CONTSAMEGRADIENTSAMECURVATURE);
END_TYPE;

TYPE IfcBSplineCurveForm = ENUMERATION OF
(POLYLINE_FORM,
CIRCULAR_ARC,
ELLIPTIC_ARC,
PARABOLIC_ARC,
HYPERBOLIC_ARC,

```
    UNSPECIFIED);  
END_TYPE;  
  
TYPE IfcSegmentIndexSelect = SELECT  
    (IfcArcIndex,  
     IfcLineIndex);  
END_TYPE;
```

```
TYPE IfcTrimmingSelect = SELECT  
    (IfcCartesianPoint,  
     IfcParameterValue);  
END_TYPE;
```

```
TYPE IfcTrimmingPreference = ENUMERATION OF  
    (CARTESIAN,  
     PARAMETER,  
     UNSPECIFIED);  
END_TYPE;
```

```
TYPE IfcProfileTypeEnum = ENUMERATION OF  
    (CURVE,  
     AREA);  
END_TYPE;
```

```
TYPE IfcActorSelect = SELECT  
    (IfcOrganization,  
     IfcPerson,  
     IfcPersonAndOrganization);  
END_TYPE;
```

TYPE IfcRoleEnum = ENUMERATION OF

(SUPPLIER,
MANUFACTURER,
CONTRACTOR,
SUBCONTRACTOR,
ARCHITECT,
STRUCTURALENGINEER,
COSTENGINEER,
CLIENT,
BUILDINGOWNER,
BUILDINGOPERATOR,
MECHANICALENGINEER,
ELECTRICALENGINEER,
PROJECTMANAGER,
FACILITIESMANAGER,
CIVILENGINEER,
COMMISSIONINGENGINEER,
ENGINEER,
OWNER,
CONSULTANT,
CONSTRUCTIONMANAGER,
FIELDCONSTRUCTIONMANAGER,
RESELLER,
USERDEFINED);

END_TYPE;

TYPE IfcURIReference = STRING;

END_TYPE;

TYPE IfcAddressTypeEnum = ENUMERATION OF

```
(OFFICE,  
SITE,  
HOME,  
DISTRIBUTIONPOINT,  
USERDEFINED);  
END_TYPE;
```

```
TYPE IfcActionRequestTypeEnum = ENUMERATION OF  
(EMAIL,  
FAX,  
PHONE,  
POST,  
VERBAL,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCostItemTypeEnum = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAppliedValueSelect = SELECT  
(IfcMeasureWithUnit,  
IfcReference,  
IfcValue);  
END_TYPE;
```

```
TYPE IfcValue = SELECT  
(IfcDerivedMeasureValue,
```

```
    IfcMeasureValue,  
    IfcSimpleValue);
```

```
END_TYPE;
```

```
TYPE IfcDerivedMeasureValue = SELECT  
  ( IfcAbsorbedDoseMeasure,  
    IfcAccelerationMeasure,  
    IfcAngularVelocityMeasure,  
    IfcAreaDensityMeasure,  
    IfcCompoundPlaneAngleMeasure,  
    IfcCurvatureMeasure,  
    IfcDoseEquivalentMeasure,  
    IfcDynamicViscosityMeasure,  
    IfcElectricCapacitanceMeasure,  
    IfcElectricChargeMeasure,  
    IfcElectricConductanceMeasure,  
    IfcElectricResistanceMeasure,  
    IfcElectricVoltageMeasure,  
    IfcEnergyMeasure,  
    IfcForceMeasure,  
    IfcFrequencyMeasure,  
    IfcHeatFluxDensityMeasure,  
    IfcHeatingValueMeasure,  
    IfcIlluminanceMeasure,  
    IfcInductanceMeasure,  
    IfcIntegerCountRateMeasure,  
    IfcIonConcentrationMeasure,  
    IfcIsothermalMoistureCapacityMeasure,  
    IfcKinematicViscosityMeasure,  
    IfcLinearForceMeasure,
```


IfcLinearMomentMeasure,
IfcLinearStiffnessMeasure,
IfcLinearVelocityMeasure,
IfcLuminousFluxMeasure,
IfcLuminousIntensityDistributionMeasure,
IfcMagneticFluxDensityMeasure,
IfcMagneticFluxMeasure,
IfcMassDensityMeasure,
IfcMassFlowRateMeasure,
IfcMassPerLengthMeasure,
IfcModulusOfElasticityMeasure,
IfcModulusOfLinearSubgradeReactionMeasure,
IfcModulusOfRotationalSubgradeReactionMeasure,
IfcModulusOfSubgradeReactionMeasure,
IfcMoistureDiffusivityMeasure,
IfcMolecularWeightMeasure,
IfcMomentOfInertiaMeasure,
IfcMonetaryMeasure,
IfcPHMeasure,
IfcPlanarForceMeasure,
IfcPowerMeasure,
IfcPressureMeasure,
IfcRadioActivityMeasure,
IfcRotationalFrequencyMeasure,
IfcRotationalMassMeasure,
IfcRotationalStiffnessMeasure,
IfcSectionModulusMeasure,
IfcSectionalAreaIntegralMeasure,
IfcShearModulusMeasure,
IfcSoundPowerLevelMeasure,

```
IfcSoundPowerMeasure,  
IfcSoundPressureLevelMeasure,  
IfcSoundPressureMeasure,  
IfcSpecificHeatCapacityMeasure,  
IfcTemperatureGradientMeasure,  
IfcTemperatureRateOfChangeMeasure,  
IfcThermalAdmittanceMeasure,  
IfcThermalConductivityMeasure,  
IfcThermalExpansionCoefficientMeasure,  
IfcThermalResistanceMeasure,  
IfcThermalTransmittanceMeasure,  
IfcTorqueMeasure,  
IfcVaporPermeabilityMeasure,  
IfcVolumetricFlowRateMeasure,  
IfcWarpingConstantMeasure,  
IfcWarpingMomentMeasure);  
END_TYPE;
```

```
TYPE IfcRadioActivityMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcRotationalFrequencyMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcRotationalMassMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcRotationalStiffnessMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSectionModulusMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSectionalAreaIntegralMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcShearModulusMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSoundPowerLevelMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSoundPowerMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSoundPressureLevelMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSoundPressureMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcSpecificHeatCapacityMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcTemperatureGradientMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcTemperatureRateOfChangeMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcThermalAdmittanceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcThermalConductivityMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcThermalExpansionCoefficientMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcThermalResistanceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcThermalTransmittanceMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcTorqueMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcVaporPermeabilityMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcVolumetricFlowRateMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcWarpingConstantMeasure = REAL;  
END_TYPE;
```

```
TYPE IfcWarpingMomentMeasure = REAL;  
END_TYPE;
```

```

TYPE IfcMeasureValue = SELECT
( IfcAmountOfSubstanceMeasure,
  IfcAreaMeasure,
  IfcComplexNumber,
  IfcContextDependentMeasure,
  IfcCountMeasure,
  IfcDescriptiveMeasure,
  IfcElectricCurrentMeasure,
  IfcLengthMeasure,
  IfcLuminousIntensityMeasure,
  IfcMassMeasure,
  IfcNonNegativeLengthMeasure,
  IfcNormalisedRatioMeasure,
  IfcNumericMeasure,
  IfcParameterValue,
  IfcPlaneAngleMeasure,
  IfcPositiveLengthMeasure,
  IfcPositivePlaneAngleMeasure,
  IfcPositiveRatioMeasure,
  IfcRatioMeasure,
  IfcSolidAngleMeasure,
  IfcThermodynamicTemperatureMeasure,
  IfcTimeMeasure,
  IfcVolumeMeasure);
END_TYPE;

```

```

TYPE IfcSolidAngleMeasure = REAL;
END_TYPE;

```

```

TYPE IfcThermodynamicTemperatureMeasure = REAL;

```

END_TYPE;

TYPE IfcTimeMeasure = REAL;

END_TYPE;

TYPE IfcVolumeMeasure = REAL;

END_TYPE;

TYPE IfcSimpleValue = SELECT

(IfcBoolean,
IfcDate,
IfcDateTime,
IfcDuration,
IfcIdentifier,
IfcInteger,
IfcLabel,
IfcLogical,
IfcPositiveInteger,
IfcReal,
IfcText,
IfcTime,
IfcTimeStamp);

END_TYPE;

TYPE IfcTime = STRING;

END_TYPE;

TYPE IfcTimeStamp = INTEGER;

END_TYPE;

```
TYPE IfcUnit = SELECT
  (IfcDerivedUnit,
   IfcMonetaryUnit,
   IfcNamedUnit);
END_TYPE;
```

```
TYPE IfcSIPrefix = ENUMERATION OF
  (EXA,
   PETA,
   TERA,
   GIGA,
   MEGA,
   KILO,
   HECTO,
   DECA,
   DECI,
   CENTI,
   MILLI,
   MICRO,
   NANO,
   PICO,
   FEMTO,
   ATTO);
END_TYPE;
```

```
TYPE IfcSIUnitName = ENUMERATION OF
  (AMPERE,
   BECQUEREL,
   CANDELA,
   COULOMB,
```

CUBIC_METRE,
DEGREE_CELSIUS,
FARAD,
GRAM,
GRAY,
HENRY,
HERTZ,
JOULE,
KELVIN,
LUMEN,
LUX,
METRE,
MOLE,
NEWTON,
OHM,
PASCAL,
RADIAN,
SECOND,
SIEMENS,
SIEVERT,
SQUARE_METRE,
STERADIAN,
TESLA,
VOLT,
WATT,
WEBER);
END_TYPE;

TYPE IfcUnitEnum = ENUMERATION OF
(ABSORBEDDOSEUNIT,

AMOUNT OF SUBSTANCE UNIT,
AREA UNIT,
DOSE EQUIVALENT UNIT,
ELECTRIC CAPACITANCE UNIT,
ELECTRIC CHARGE UNIT,
ELECTRIC CONDUCTANCE UNIT,
ELECTRIC CURRENT UNIT,
ELECTRIC RESISTANCE UNIT,
ELECTRIC VOLTAGE UNIT,
ENERGY UNIT,
FORCE UNIT,
FREQUENCY UNIT,
ILLUMINANCE UNIT,
INDUCTANCE UNIT,
LENGTH UNIT,
LUMINOUS FLUX UNIT,
LUMINOUS INTENSITY UNIT,
MAGNETIC FLUX DENSITY UNIT,
MAGNETIC FLUX UNIT,
MASS UNIT,
PLANE ANGLE UNIT,
POWER UNIT,
PRESSURE UNIT,
RADIOACTIVITY UNIT,
SOLID ANGLE UNIT,
THERMODYNAMIC TEMPERATURE UNIT,
TIME UNIT,
VOLUME UNIT,
USER DEFINED);

END_TYPE;

TYPE IfcDerivedUnitEnum = ENUMERATION OF

(ANGULARVELOCITYUNIT,
AREADENSITYUNIT,
COMPOUNDPLANEANGLEUNIT,
DYNAMICVISCOSITYUNIT,
HEATFLUXDENSITYUNIT,
INTEGERCOUNTRATEUNIT,
ISOTHERMALMOISTURECAPACITYUNIT,
KINEMATICVISCOSITYUNIT,
LINEARVELOCITYUNIT,
MASSDENSITYUNIT,
MASSFLOWRATEUNIT,
MOISTUREDIFFUSIVITYUNIT,
MOLECULARWEIGHTUNIT,
SPECIFICHEATCAPACITYUNIT,
THERMALADMITTANCEUNIT,
THERMALCONDUCTANCEUNIT,
THERMALRESISTANCEUNIT,
THERMALTRANSMITTANCEUNIT,
VAPORPERMEABILITYUNIT,
VOLUMETRICFLOWRATEUNIT,
ROTATIONALFREQUENCYUNIT,
TORQUEUNIT,
MOMENTOFINERTIAUNIT,
LINEARMOMENTUNIT,
LINEARFORCEUNIT,
PLANARFORCEUNIT,
MODULUSOFELASTICITYUNIT,
SHEARMODULUSUNIT,

LINEARSTIFFNESSUNIT,
ROTATIONALSTIFFNESSUNIT,
MODULUSOF SUBGRADEREACTIONUNIT,
ACCELERATIONUNIT,
CURVATUREUNIT,
HEATINGVALUEUNIT,
IONCONCENTRATIONUNIT,
LUMINOUSINTENSITYDISTRIBUTIONUNIT,
MASSPERLENGTHUNIT,
MODULUSOF LINEAR SUBGRADEREACTIONUNIT,
MODULUSOF ROTATIONAL SUBGRADEREACTIONUNIT,
PHUNIT,
ROTATIONALMASSUNIT,
SECTIONAREA INTEGRALUNIT,
SECTIONMODULUSUNIT,
SOUNDPOWERLEVELUNIT,
SOUNDPOWERUNIT,
SOUNDPRESSURELEVELUNIT,
SOUNDPRESSUREUNIT,
TEMPERATUREGRADIENTUNIT,
TEMPERATURERATEOFCHANGEUNIT,
THERMALEXPANSIONCOEFFICIENTUNIT,
WARPINGCONSTANTUNIT,
WARPINGMOMENTUNIT,
USERDEFINED);
END_TYPE;

TYPE IfcArithmeticOperatorEnum = ENUMERATION OF
(ADD,
DIVIDE,

```
MULTIPLY,  
SUBTRACT);  
END_TYPE;
```

```
TYPE IfcCostScheduleTypeEnum = ENUMERATION OF  
(BUDGET,  
COSTPLAN,  
ESTIMATE,  
TENDER,  
PRICEDBILLOFQUANTITIES,  
UNPRICEDBILLOFQUANTITIES,  
SCHEDULEOFRATES,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPerformanceHistoryTypeEnum = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPermitTypeEnum = ENUMERATION OF  
(ACCESS,  
BUILDING,  
WORK,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcProjectOrderTypeEnum = ENUMERATION OF
```

```
(CHANGEORDER,  
  MAINTENANCEWORKORDER,  
  MOVEORDER,  
  PURCHASEORDER,  
  WORKORDER,  
  USERDEFINED,  
  NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTimeOrRatioSelect = SELECT  
  (IfcDuration,  
   IfcRatioMeasure);  
END_TYPE;
```

```
TYPE IfcTaskDurationEnum = ENUMERATION OF  
  (ELAPSED TIME,  
   WORK TIME,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcRecurrenceTypeEnum = ENUMERATION OF  
  (DAILY,  
   WEEKLY,  
   MONTHLY_BY_DAY_OF_MONTH,  
   MONTHLY_BY_POSITION,  
   BY_DAY_COUNT,  
   BY_WEEKDAY_COUNT,  
   YEARLY_BY_DAY_OF_MONTH,  
   YEARLY_BY_POSITION);  
END_TYPE;
```

TYPE IfcDataOriginEnum = ENUMERATION OF

(MEASURED,
PREDICTED,
SIMULATED,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcWorkCalendarTypeEnum = ENUMERATION OF

(FIRSTSHIFT,
SECONDSHIFT,
THIRDSHIFT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcWorkPlanTypeEnum = ENUMERATION OF

(ACTUAL,
BASELINE,
PLANNED,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcWorkScheduleTypeEnum = ENUMERATION OF

(ACTUAL,
BASELINE,
PLANNED,
USERDEFINED,

```
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcInventoryTypeEnum = ENUMERATION OF  
    (ASSETINVENTORY,  
     SPACEINVENTORY,  
     FURNITUREINVENTORY,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcLoadGroupTypeEnum = ENUMERATION OF  
    (LOAD_GROUP,  
     LOAD_CASE,  
     LOAD_COMBINATION,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcActionTypeEnum = ENUMERATION OF  
    (PERMANENT_G,  
     VARIABLE_Q,  
     EXTRAORDINARY_A,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcActionSourceTypeEnum = ENUMERATION OF  
    (DEAD_LOAD_G,  
     COMPLETION_G1,
```

```
LIVE_LOAD_Q,  
SNOW_S,  
WIND_W,  
PRESTRESSING_P,  
SETTLEMENT_U,  
TEMPERATURE_T,  
EARTHQUAKE_E,  
FIRE,  
IMPULSE,  
IMPACT,  
TRANSPORT,  
ERECTION,  
PROPPING,  
SYSTEM_IMPERFECTION,  
SHRINKAGE,  
CREEP,  
LACK_OF_FIT,  
BUOYANCY,  
ICE,  
CURRENT,  
WAVE,  
RAIN,  
BRAKES,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAnalysisTheoryTypeEnum = ENUMERATION OF  
(FIRST_ORDER_THEORY,  
SECOND_ORDER_THEORY,
```



```
THIRD_ORDER_THEORY,  
FULL_NONLINEAR_THEORY,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBuildingSystemTypeEnum = ENUMERATION OF  
(FENESTRATION,  
FOUNDATION,  
LOADBEARING,  
OUTERSHELL,  
SHADING,  
TRANSPORT,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDistributionSystemEnum = ENUMERATION OF  
(AIRCONDITIONING,  
AUDIOVISUAL,  
CHEMICAL,  
CHILLEDWATER,  
COMMUNICATION,  
COMPRESSED AIR,  
CONDENSERWATER,  
CONTROL,  
CONVEYING,  
DATA,  
DISPOSAL,  
DOMESTIC COLDWATER,
```

DOMESTIC HOT WATER,
DRAINAGE,
EARTHING,
ELECTRICAL,
ELECTROACOUSTIC,
EXHAUST,
FIRE PROTECTION,
FUEL,
GAS,
HAZARDOUS,
HEATING,
LIGHTING,
LIGHTNING PROTECTION,
MUNICIPAL SOLID WASTE,
OIL,
OPERATIONAL,
POWER GENERATION,
RAIN WATER,
REFRIGERATION,
SECURITY,
SEWAGE,
SIGNAL,
STORM WATER,
TELEPHONE,
TV,
VACUUM,
VENT,
VENTILATION,
WASTEWATER,
WATER SUPPLY,

```
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcConnectionTypeEnum = ENUMERATION OF  
(ATPATH,  
ATSTART,  
ATEND,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcExternalSpatialElementTypeEnum = ENUMERATION OF  
(EXTERNAL,  
EXTERNAL_EARTH,  
EXTERNAL_WATER,  
EXTERNAL_FIRE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCurveOrEdgeCurve = SELECT  
(IfcBoundedCurve,  
IfcEdgeCurve);  
END_TYPE;
```

```
TYPE IfcCurveStyleFontSelect = SELECT  
(IfcCurveStyleFont,  
IfcPreDefinedCurveFont);  
END_TYPE;
```

```
TYPE IfcTextFontName = STRING;  
END_TYPE;
```

```
TYPE IfcSizeSelect = SELECT  
  (IfcDescriptiveMeasure,  
   IfcLengthMeasure,  
   IfcNormalisedRatioMeasure,  
   IfcPositiveLengthMeasure,  
   IfcPositiveRatioMeasure,  
   IfcRatioMeasure);  
END_TYPE;
```

```
TYPE IfcColourOrFactor = SELECT  
  (IfcColourRgb,  
   IfcNormalisedRatioMeasure);  
END_TYPE;
```

```
TYPE IfcSpecularHighlightSelect = SELECT  
  (IfcSpecularExponent,  
   IfcSpecularRoughness);  
END_TYPE;
```

```
TYPE IfcSpecularExponent = REAL;  
END_TYPE;
```

```
TYPE IfcSpecularRoughness = REAL;  
  WHERE  
    WR1 : {0.0 <= SELF <= 1.0};  
END_TYPE;
```

TYPE IfcReflectanceMethodEnum = ENUMERATION OF

(BLINN,
FLAT,
GLASS,
MATT,
METAL,
MIRROR,
PHONG,
PLASTIC,
STRAUSS,
NOTDEFINED);

END_TYPE;

TYPE IfcColour = SELECT

(IfcColourSpecification,
IfcPreDefinedColour);

END_TYPE;

TYPE IfcTextAlignment = STRING;

WHERE

WR1 : SELF IN ['left', 'right', 'center', 'justify'];

END_TYPE;

TYPE IfcTextDecoration = STRING;

WHERE

WR1 : SELF IN ['none', 'underline', 'overline', 'line-through', 'blink'];

END_TYPE;

TYPE IfcTextTransformation = STRING;

WHERE

```
WR1 : SELF IN ['capitalize', 'uppercase', 'lowercase', 'none'];  
END_TYPE;
```

```
TYPE IfcPointOrVertexPoint = SELECT  
  (IfcPoint,  
   IfcVertexPoint);  
END_TYPE;
```

```
TYPE IfcSurfaceOrFaceSurface = SELECT  
  (IfcFaceBasedSurfaceModel,  
   IfcFaceSurface,  
   IfcSurface);  
END_TYPE;
```

```
TYPE IfcSolidOrShell = SELECT  
  (IfcClosedShell,  
   IfcSolidModel);  
END_TYPE;
```

```
TYPE IfcCsgSelect = SELECT  
  (IfcBooleanResult,  
   IfcCsgPrimitive3D);  
END_TYPE;
```

```
TYPE IfcBooleanOperator = ENUMERATION OF  
  (UNION,  
   INTERSECTION,  
   DIFFERENCE);  
END_TYPE;
```

TYPE IfcBooleanOperand = SELECT

(IfcBooleanResult,
IfcCsgPrimitive3D,
IfcHalfSpaceSolid,
IfcSolidModel);

END_TYPE;

TYPE IfcPhysicalOrVirtualEnum = ENUMERATION OF

(PHYSICAL,
VIRTUAL,
NOTDEFINED);

END_TYPE;

TYPE IfcInternalOrExternalEnum = ENUMERATION OF

(INTERNAL,
EXTERNAL,
EXTERNAL_EARTH,
EXTERNAL_WATER,
EXTERNAL_FIRE,
NOTDEFINED);

END_TYPE;

TYPE IfcSpaceBoundarySelect = SELECT

(IfcExternalSpatialElement,
IfcSpace);

END_TYPE;

TYPE IfcElementCompositionEnum = ENUMERATION OF

(COMPLEX,
ELEMENT,

PARTIAL);
END_TYPE;

TYPE IfcSpaceTypeEnum = ENUMERATION OF

(SPACE,
PARKING,
GFA,
INTERNAL,
EXTERNAL,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcBeamTypeEnum = ENUMERATION OF

(BEAM,
JOIST,
HOLLOWCORE,
L INTEL,
SPANDREL,
T_BEAM,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcBuildingElementProxyTypeEnum = ENUMERATION OF

(COMPLEX,
ELEMENT,
PARTIAL,
PROVISIONFORVOID,
USERDEFINED,


```
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcChimneyTypeEnum = ENUMERATION OF  
    (USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcColumnTypeEnum = ENUMERATION OF  
    (COLUMN,  
     PILASTER,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCurtainWallTypeEnum = ENUMERATION OF  
    (USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorTypeEnum = ENUMERATION OF  
    (DOOR,  
     GATE,  
     TRAPDOOR,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorTypeOperationEnum = ENUMERATION OF  
    (SINGLE_SWING_LEFT,
```

```
SINGLE_SWING_RIGHT,  
DOUBLE_DOOR_SINGLE_SWING,  
DOUBLE_DOOR_SINGLE_SWING_OPPOSITE_LEFT,  
DOUBLE_DOOR_SINGLE_SWING_OPPOSITE_RIGHT,  
DOUBLE_SWING_LEFT,  
DOUBLE_SWING_RIGHT,  
DOUBLE_DOOR_DOUBLE_SWING,  
SLIDING_TO_LEFT,  
SLIDING_TO_RIGHT,  
DOUBLE_DOOR_SLIDING,  
FOLDING_TO_LEFT,  
FOLDING_TO_RIGHT,  
DOUBLE_DOOR_FOLDING,  
REVOLVING,  
ROLLINGUP,  
SWING_FIXED_LEFT,  
SWING_FIXED_RIGHT,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcFootingTypeEnum = ENUMERATION OF  
(CAISSON_FOUNDATION,  
FOOTING_BEAM,  
PAD_FOOTING,  
PILE_CAP,  
STRIP_FOOTING,  
BASE_FOOTING_K,  
PITMOUTH_FOOTING_K,  
PSC_PILE_FOUNDATION_K,
```

```
RC_PILE_FOUNDATION_K,  
SLURRY_WALL_FOUNDATION_K,  
SPREAD_FOUNDATION_K,  
STEEL_PILE_FOUNDATION_K,  
STEEL_SHEET_PILE_FOUNDATION_K,  
MAT_FOUNDATION_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcMemberTypeEnum = ENUMERATION OF
```

```
(BRACE,  
CHORD,  
COLLAR,  
MEMBER,  
MULLION,  
PLATE,  
POST,  
PURLIN,  
RAFTER,  
STRINGER,  
STRUT,  
STUD,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPileTypeEnum = ENUMERATION OF
```

```
(BORED,  
DRIVEN,
```

```
JETGROUTING,  
COHESION,  
FRICTION,  
SUPPORT,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPileConstructionEnum = ENUMERATION OF  
(CAST_IN_PLACE,  
COMPOSITE,  
PRECAST_CONCRETE,  
PREFAB_STEEL,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPlateTypeEnum = ENUMERATION OF  
(CURTAIN_PANEL,  
SHEET,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcRailingTypeEnum = ENUMERATION OF  
(HANDRAIL,  
GUARDRAIL,  
BALUSTRADE,  
USERDEFINED,  
NOTDEFINED);
```

END_TYPE;

TYPE IfcRampTypeEnum = ENUMERATION OF

(STRAIGHT_RUN_RAMP,
TWO_STRAIGHT_RUN_RAMP,
QUARTER_TURN_RAMP,
TWO_QUARTER_TURN_RAMP,
HALF_TURN_RAMP,
SPIRAL_RAMP,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcRampFlightTypeEnum = ENUMERATION OF

(STRAIGHT,
SPIRAL,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcRoofTypeEnum = ENUMERATION OF

(FLAT_ROOF,
SHED_ROOF,
GABLE_ROOF,
HIP_ROOF,
HIPPED_GABLE_ROOF,
GAMBREL_ROOF,
MANSARD_ROOF,
BARREL_ROOF,
RAINBOW_ROOF,

```
BUTTERFLY_ROOF,  
PAVILION_ROOF,  
DOME_ROOF,  
FREEFORM,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcShadingDeviceTypeEnum = ENUMERATION OF  
(JALOUSIE,  
SHUTTER,  
AWNING,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSlabTypeEnum = ENUMERATION OF  
(FLOOR,  
ROOF,  
LANDING,  
BASESLAB,  
INVERT_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcStairTypeEnum = ENUMERATION OF  
(STRAIGHT_RUN_STAIR,  
TWO_STRAIGHT_RUN_STAIR,  
QUARTER_WINDING_STAIR,
```

```
QUARTER_TURN_STAIR,  
HALF_WINDING_STAIR,  
HALF_TURN_STAIR,  
TWO_QUARTER_WINDING_STAIR,  
TWO_QUARTER_TURN_STAIR,  
THREE_QUARTER_WINDING_STAIR,  
THREE_QUARTER_TURN_STAIR,  
SPIRAL_STAIR,  
DOUBLE_RETURN_STAIR,  
CURVED_RUN_STAIR,  
TWO_CURVED_RUN_STAIR,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcStairFlightTypeEnum = ENUMERATION OF  
(STRAIGHT,  
WINDER,  
SPIRAL,  
CURVED,  
FREEFORM,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcWallTypeEnum = ENUMERATION OF  
(MOVABLE,  
PARAPET,  
PARTITIONING,  
PLUMBINGWALL,
```

```
SHEAR,  
SOLIDWALL,  
STANDARD,  
POLYGONAL,  
ELEMENTEDWALL,  
WINGWALL_K,  
CHESTWALL_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcWindowTypeEnum = ENUMERATION OF
```

```
(WINDOW,  
SKYLIGHT,  
LIGHTDOME,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcWindowTypePartitioningEnum = ENUMERATION OF
```

```
(SINGLE_PANEL,  
DOUBLE_PANEL_VERTICAL,  
DOUBLE_PANEL_HORIZONTAL,  
TRIPLE_PANEL_VERTICAL,  
TRIPLE_PANEL_BOTTOM,  
TRIPLE_PANEL_TOP,  
TRIPLE_PANEL_LEFT,  
TRIPLE_PANEL_RIGHT,  
TRIPLE_PANEL_HORIZONTAL,  
USERDEFINED,
```


NOTDEFINED);
END_TYPE;

TYPE IfcCoveringTypeEnum = ENUMERATION OF

(CEILING,
FLOORING,
CLADDING,
ROOFING,
MOLDING,
SKIRTINGBOARD,
INSULATION,
MEMBRANE,
SLEEVING,
WRAPPING,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcSpatialZoneTypeEnum = ENUMERATION OF

(CONSTRUCTION,
FIRESAFETY,
LIGHTING,
OCCUPANCY,
SECURITY,
THERMAL,
TRANSPORT,
VENTILATION,
CIVIL_STRUCTURE_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcRoadTypeEnum_K = ENUMERATION OF

(RAMPROAD_K,
APPROACHROAD_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcBridgeTypeEnum_K = ENUMERATION OF

(RC_SLAB_BRIDGE_K,
RC_HOLLOW_SLAB_BRIDGE_K,
RC_T_BEAM_GIRDER_BRIDGE_K,
STEEL_I_BEAM_GIRDER_BRIDGE_K,
PSC_BOX_GIRDER_BRIDGE_K,
STEEL_PLATE_GIRDER_BRIDGE_K,
RAHMEN_BRIDGE_K,
TRUSS_BRIDGE_K,
ARCH_BRIDGE_K,
CABLE_STAYED_BRIDGE_K,
SUSPENSION_BRIDGE_K,
PREFLEX_GIRDER_BRIDGE_K,
PSC_I_GIRDER_BRIDGE_K,
PSC_SLAB_BRIDGE_K,
PSC_HOLLOW_SLAB_BRIDGE_K,
RC_BOX_GIRDDER_BRIDGE_K,
STEEL_BOX_GIRDDER_BRIDGE_K,
OVERPASS_BRIDGE_K,
RAMP_BRIDGE_K,
APPROACH_BRIDGE_K,

```
EXTRADOSED_BRIDGE_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTunnelTypeEnum_K = ENUMERATION OF  
(NATM_K,  
TBM_SHIELD_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcLinearRefSpaceTypeEnum_K = ENUMERATION OF  
(FRONTAGEROAD_K,  
ROADFOOTPATH_K,  
ROADMARGINALSTRIP_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCurvilinearNodeSpaceTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcVerticalSubspaceTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

TYPE IfcGridTypeEnum = ENUMERATION OF

(RECTANGULAR,
RADIAL,
TRIANGULAR,
IRREGULAR,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcFlowDirectionEnum = ENUMERATION OF

(SOURCE,
SINK,
SOURCEANDSINK,
NOTDEFINED);

END_TYPE;

TYPE IfcDistributionPortTypeEnum = ENUMERATION OF

(CABLE,
CABLECARRIER,
DUCT,
PIPE,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcActuatorTypeEnum = ENUMERATION OF

(ELECTRICA CTUATOR,
HANDOPERATEDACTUATOR,
HYDRAULICA CTUATOR,
PNEUMATICA CTUATOR,

```
THERMOSTATICACTUATOR,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAlarmTypeEnum = ENUMERATION OF
```

```
(BELL,  
BREAKGLASSBUTTON,  
LIGHT,  
MANUALPULLBOX,  
SIREN,  
WHISTLE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcControllerTypeEnum = ENUMERATION OF
```

```
(FLOATING,  
PROGRAMMABLE,  
PROPORTIONAL,  
MULTIPOSITION,  
TWOPOSITION,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcFlowInstrumentTypeEnum = ENUMERATION OF
```

```
(PRESSUREGAUGE,  
THERMOMETER,  
AMMETER,
```

```
FREQUENCYMETER,  
POWERFACTORMETER,  
PHASEANGLEMETER,  
VOLTMETER_PEAK,  
VOLTMETER_RMS,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcProtectiveDeviceTrippingUnitTypeEnum = ENUMERATION OF  
(ELECTRONIC,  
ELECTROMAGNETIC,  
RESIDUALCURRENT,  
THERMAL,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSensorTypeEnum = ENUMERATION OF  
(CO2SENSOR,  
CONDUCTANCESENSOR,  
CONTACTSENSOR,  
FIRESSENSOR,  
FLOWSSENSOR,  
FROSTSENSOR,  
GASSENSOR,  
HEATSENSOR,  
HUMIDITYSENSOR,  
IDENTIFIERSENSOR,  
IONCONCENTRATIONSENSOR,
```

LEVELSENSOR,
LIGHTSENSOR,
MOISTURESENSOR,
MOVEMENTSENSOR,
PHSENSOR,
PRESSURESENSOR,
RADIATIONSENSOR,
RADIOACTIVITYSENSOR,
SMOKESENSOR,
SOUNDSENSOR,
TEMPERATURESENSOR,
WINDSENSOR,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcUnitaryControlElementTypeEnum = ENUMERATION OF

(ALARMPANEL,
CONTROLPANEL,
GASDETECTIONPANEL,
INDICATORPANEL,
MIMICPANEL,
HUMIDISTAT,
THERMOSTAT,
WEATHERSTATION,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcDistributionChamberElementTypeEnum = ENUMERATION OF

```
(FORMEDDUCT,  
INSPECTIONCHAMBER,  
INSPECTIONPIT,  
MANHOLE,  
METERCHAMBER,  
SUMP,  
TRENCH,  
VALVECHAMBER,  
COLLECTINGWELL_K,  
CATCHDRAIN_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAirToAirHeatRecoveryTypeEnum = ENUMERATION OF
```

```
(FIXEDPLATECOUNTERFLOWEXCHANGER,  
FIXEDPLATECROSSFLOWEXCHANGER,  
FIXEDPLATEPARALLELFLOWEXCHANGER,  
ROTARYWHEEL,  
RUNAROUNDCOILLOOP,  
HEATPIPE,  
TWINTOWERENTHALPYRECOVERYLOOPS,  
THERMOSIPHONSEALEDTUBEHEATEXCHANGERS,  
THERMOSIPHONCOILTYPEHEATEXCHANGERS,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBoilerTypeEnum = ENUMERATION OF
```

```
(WATER,
```



```
    STEAM,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBurnerTypeEnum = ENUMERATION OF  
    (USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcChillerTypeEnum = ENUMERATION OF  
    (AIRCOOLED,  
    WATERCOOLED,  
    HEATRECOVERY,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCoilTypeEnum = ENUMERATION OF  
    (DXCOOLINGCOIL,  
    ELECTRICHEATINGCOIL,  
    GASHEATINGCOIL,  
    HYDRONICCOIL,  
    STEAMHEATINGCOIL,  
    WATERCOOLINGCOIL,  
    WATERHEATINGCOIL,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

TYPE IfcCondenserTypeEnum = ENUMERATION OF

(AIRCOOLED,
EVAPORATIVECOOLED,
WATERCOOLED,
WATERCOOLEDBRAZEDPLATE,
WATERCOOLEDSHELLCOIL,
WATERCOOLEDSHELLTUBE,
WATERCOOLEDTUBEINTUBE,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcCooledBeamTypeEnum = ENUMERATION OF

(ACTIVE,
PASSIVE,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcCoolingTowerTypeEnum = ENUMERATION OF

(NATURALDRAFT,
MECHANICALINDUCEDDRAFT,
MECHANICALFORCEDDRAFT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcElectricGeneratorTypeEnum = ENUMERATION OF

(CHP,
ENGINEGENERATOR,

```
STANDALONE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcElectricMotorTypeEnum = ENUMERATION OF  
(DC,  
INDUCTION,  
POLYPHASE,  
RELUCTANCESYNCHRONOUS,  
SYNCHRONOUS,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcEngineTypeEnum = ENUMERATION OF  
(EXTERNALCOMBUSTION,  
INTERNALCOMBUSTION,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcEvaporativeCoolerTypeEnum = ENUMERATION OF  
(DIRECTEVAPORATIVERANDOMMEDIAAIRCOOLER,  
DIRECTEVAPORATIVERIGIDMEDIAAIRCOOLER,  
DIRECTEVAPORATIVESLINGERSPACKAGEDAIRCOOLER,  
DIRECTEVAPORATIVEPACKAGEDROTARYAIRCOOLER,  
DIRECTEVAPORATIVEAIRWASHER,  
INDIRECTEVAPORATIVEPACKAGEAIRCOOLER,  
INDIRECTEVAPORATIVEWETCOIL,
```

```
    INDIRECTEVAPORATIVECOOLINGTOWERORCOILCOOLER,  
    INDIRECTDIRECTCOMBINATION,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcEvaporatorTypeEnum = ENUMERATION OF  
    (DIRECTEXPANSION,  
    DIRECTEXPANSIONSHELLANDTUBE,  
    DIRECTEXPANSIONTUBEINTUBE,  
    DIRECTEXPANSIONBRAZEDPLATE,  
    FLOODEDSHELLANDTUBE,  
    SHELLANDCOIL,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcHeatExchangerTypeEnum = ENUMERATION OF  
    (PLATE,  
    SHELLANDTUBE,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcHumidifierTypeEnum = ENUMERATION OF  
    (STEAMINJECTION,  
    ADIABATICAIRWASHER,  
    ADIABATICPAN,  
    ADIABATICWETTEDELEMENT,  
    ADIABATICATOMIZING,
```

```
ADIABATICULTRASONIC,  
ADIABATICRIGIDMEDIA,  
ADIABATICCOMPRESSED AIRNOZZLE,  
ASSISTEDELECTRIC,  
ASSISTEDNATURALGAS,  
ASSISTEDPROPANE,  
ASSISTEDBUTANE,  
ASSISTEDSTEAM,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcMotorConnectionTypeEnum = ENUMERATION OF  
(BELTDRIVE,  
COUPLING,  
DIRECTDRIVE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSolarDeviceTypeEnum = ENUMERATION OF  
(SOLARCOLLECTOR,  
SOLARPANEL,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTransformerTypeEnum = ENUMERATION OF  
(CURRENT,  
FREQUENCY,
```

```
    INVERTER,  
    RECTIFIER,  
    VOLTAGE,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTubeBundleTypeEnum = ENUMERATION OF  
    (FINNED,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcUnitaryEquipmentTypeEnum = ENUMERATION OF  
    (AIRHANDLER,  
    AIRCONDITIONINGUNIT,  
    DEHUMIDIFIER,  
    SPLITSYSTEM,  
    ROOFTOPUNIT,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAirTerminalBoxTypeEnum = ENUMERATION OF  
    (CONSTANTFLOW,  
    VARIABLEFLOWPRESSUREDEPENDANT,  
    VARIABLEFLOWPRESSUREINDEPENDANT,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

TYPE IfcDamperTypeEnum = ENUMERATION OF

(BACKDRAFTDAMPER,
BALANCINGDAMPER,
BLASTDAMPER,
CONTROLDAMPER,
FIREDAMPER,
FIRESMOKEDAMPER,
FUMEHOODEXHAUST,
GRAVITYDAMPER,
GRAVITYRELIEFDAMPER,
RELIEFDAMPER,
SMOKEDAMPER,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcElectricDistributionBoardTypeEnum = ENUMERATION OF

(CONSUMERUNIT,
DISTRIBUTIONBOARD,
MOTORCONTROLCENTRE,
SWITCHBOARD,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcElectricTimeControlTypeEnum = ENUMERATION OF

(TIMECLOCK,
TIMEDELAY,
RELAY,

```
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcFlowMeterTypeEnum = ENUMERATION OF  
(ENERGYMETER,  
GASMETER,  
OILMETER,  
WATERMETER,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcProtectiveDeviceTypeEnum = ENUMERATION OF  
(CIRCUITBREAKER,  
EARTHLEAKAGECIRCUITBREAKER,  
EARTHINGSWITCH,  
FUSEDISCONNECTOR,  
RESIDUALCURRENTCIRCUITBREAKER,  
RESIDUALCURRENTSWITCH,  
VARISTOR,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSwitchingDeviceTypeEnum = ENUMERATION OF  
(CONTACTOR,  
DIMMERSWITCH,  
EMERGENCYSTOP,  
KEYPAD,
```


MOMENTARYSWITCH,
SELECTORSWITCH,
STARTER,
SWITCHDISCONNECTOR,
TOGGLESWITCH,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcValveTypeEnum = ENUMERATION OF

(AIRRELEASE,
ANTIVACUUM,
CHANGEOVER,
CHECK,
COMMISSIONING,
DIVERTING,
DRAWOFFCOCK,
DOUBLECHECK,
DOUBLEREGULATING,
FAUCET,
FLUSHING,
GASCOCK,
GASTAP,
ISOLATING,
MIXING,
PRESSUREREDUCING,
PRESSURERELIEF,
REGULATING,
SAFETYCUTOFF,
STEAMTRAP,

```
    STOPCOCK,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCableCarrierFittingTypeEnum = ENUMERATION OF  
    (BEND,  
    CROSS,  
    REDUCER,  
    TEE,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCableFittingTypeEnum = ENUMERATION OF  
    (CONNECTOR,  
    ENTRY,  
    EXIT,  
    JUNCTION,  
    TRANSITION,  
    USERDEFINED,  
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDuctFittingTypeEnum = ENUMERATION OF  
    (BEND,  
    CONNECTOR,  
    ENTRY,  
    EXIT,  
    JUNCTION,
```

```
OBSTRUCTION,  
TRANSITION,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcJunctionBoxTypeEnum = ENUMERATION OF  
(DATA,  
POWER,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPipeFittingTypeEnum = ENUMERATION OF  
(BEND,  
CONNECTOR,  
ENTRY,  
EXIT,  
JUNCTION,  
OBSTRUCTION,  
TRANSITION,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcGutterFittingTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

TYPE IfcCompressorTypeEnum = ENUMERATION OF

(DYNAMIC,
RECIPROCATING,
ROTARY,
SCROLL,
TROCHOIDAL,
SINGLESTAGE,
BOOSTER,
OPENTYPE,
HERMETIC,
SEMIHERMETIC,
WELDEDSHELLHERMETIC,
ROLLINGPISTON,
ROTARYVANE,
SINGLESCREW,
TWINSCREW,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcFanTypeEnum = ENUMERATION OF

(CENTRIFUGALFORWARDCURVED,
CENTRIFUGALRADIAL,
CENTRIFUGALBACKWARDINCLINEDCURVED,
CENTRIFUGALAIRFOIL,
TUBEAXIAL,
VANEAXIAL,
PROPELLORAXIAL,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcPumpTypeEnum = ENUMERATION OF

(CIRCULATOR,
ENDSUCTION,
SPLITCASE,
SUBMERSIBLEPUMP,
SUMPUMP,
VERTICALINLINE,
VERTICALTURBINE,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcCableCarrierSegmentTypeEnum = ENUMERATION OF

(CABLELADDERSEGMENT,
CABLETRAYSEGMENT,
CABLETRUNKINGSEGMENT,
CONDUITSEGMENT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcCableSegmentTypeEnum = ENUMERATION OF

(BUSBARSEGMENT,
CABLESEGMENT,
CONDUCTORSEGMENT,
CORESEGMENT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcDuctSegmentTypeEnum = ENUMERATION OF

(RIGIDSEGMENT,
FLEXIBLESEGMENT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcPipeSegmentTypeEnum = ENUMERATION OF

(CULVERT,
FLEXIBLESEGMENT,
RIGIDSEGMENT,
GUTTER,
SPOOL,
CLOSEDCONDUIT_K,
PERFORATED_K,
HUMEGMENT_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcGutterSegmentTypeEnum_K = ENUMERATION OF

(V_TYPE_K,
L_TYPE_K,
U_TYPE_K,
CANAL_TYPE_K,
OPENCHANNEL_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcElectricFlowStorageDeviceTypeEnum = ENUMERATION OF

(BATTERY,
CAPACITORBANK,
HARMONICFILTER,
INDUCTORBANK,
UPS,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcTankTypeEnum = ENUMERATION OF

(BASIN,
BREAKPRESSURE,
EXPANSION,
FEEDANDEXPANSION,
PRESSUREVESSEL,
STORAGE,
VESSEL,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcAirTerminalTypeEnum = ENUMERATION OF

(DIFFUSER,
GRILLE,
LOUVRE,
REGISTER,
USERDEFINED,

NOTDEFINED);
END_TYPE;

TYPE IfcAudioVisualApplianceTypeEnum = ENUMERATION OF
(AMPLIFIER,
CAMERA,
DISPLAY,
MICROPHONE,
PLAYER,
PROJECTOR,
RECEIVER,
SPEAKER,
SWITCHER,
TELEPHONE,
TUNER,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcCommunicationsApplianceTypeEnum = ENUMERATION OF
(ANTENNA,
COMPUTER,
FAX,
GATEWAY,
MODEM,
NETWORKAPPLIANCE,
NETWORKBRIDGE,
NETWORKHUB,
PRINTER,
REPEATER,


```
ROUTER,  
SCANNER,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcElectricApplianceTypeEnum = ENUMERATION OF
```

```
(DISHWASHER,  
ELECTRICOOKER,  
FREESTANDINGELECTRICHEATER,  
FREESTANDINGFAN,  
FREESTANDINGWATERHEATER,  
FREESTANDINGWATERCOOLER,  
FREEZER,  
FRIDGE_FREEZER,  
HANDDRYER,  
KITCHENMACHINE,  
MICROWAVE,  
PHOTOCOPIER,  
REFRIGERATOR,  
TUMBLEDRYER,  
VENDINGMACHINE,  
WASHINGMACHINE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcFireSuppressionTerminalTypeEnum = ENUMERATION OF
```

```
(BREECHINGINLET,  
FIREHYDRANT,
```

```
HOSEREEL,  
SPRINKLER,  
SPRINKLERDEFLECTOR,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcLampTypeEnum = ENUMERATION OF  
(COMPACTFLUORESCENT,  
FLUORESCENT,  
HALOGEN,  
HIGHPRESSUREMERCURY,  
HIGHPRESSURESODIUM,  
LED,  
METALHALIDE,  
OLED,  
TUNGSTENFILAMENT,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcLightFixtureTypeEnum = ENUMERATION OF  
(POINTSOURCE,  
DIRECTIONSOURCE,  
SECURITYLIGHTING,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcMedicalDeviceTypeEnum = ENUMERATION OF
```

```
(AIRSTATION,  
FEEDAIRUNIT,  
OXYGENGENERATOR,  
OXYGENPLANT,  
VACUUMSTATION,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcOutletTypeEnum = ENUMERATION OF
```

```
(AUDIOVISUALOUTLET,  
COMMUNICATIONSOUTLET,  
POWEROUTLET,  
DATAOUTLET,  
TELEPHONEOUTLET,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSanitaryTerminalTypeEnum = ENUMERATION OF
```

```
(BATH,  
BIDET,  
CISTERN,  
SHOWER,  
SINK,  
SANITARYFOUNTAIN,  
TOILETPAN,  
URINAL,  
WASHHANDBASIN,  
WCSEAT,
```

```
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSpaceHeaterTypeEnum = ENUMERATION OF  
(CONVECTOR,  
RADIATOR,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcStackTerminalTypeEnum = ENUMERATION OF  
(BIRDCAGE,  
COWL,  
RAINWATERHOPPER,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcWasteTerminalTypeEnum = ENUMERATION OF  
(FLOORTRAP,  
FLOORWASTE,  
GULLYSUMP,  
GULLYTRAP,  
ROOFDRAIN,  
WASTEDISPOSALUNIT,  
WASTETRAP,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

TYPE IfcDuctSilencerTypeEnum = ENUMERATION OF

(FLATOVAL,
RECTANGULAR,
ROUND,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcFilterTypeEnum = ENUMERATION OF

(AIRPARTICLEFILTER,
COMPRESSED AIRFILTER,
ODORFILTER,
OILFILTER,
STRAINER,
WATERFILTER,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcInterceptorTypeEnum = ENUMERATION OF

(CYCLONIC,
GREASE,
OIL,
PETROL,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcObjectTypeEnum = ENUMERATION OF

```
(PRODUCT,  
PROCESS,  
CONTROL,  
RESOURCE,  
ACTOR,  
GROUP,  
PROJECT,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcProjectedOrTrueLengthEnum = ENUMERATION OF  
(PROJECTED_LENGTH,  
TRUE_LENGTH);  
END_TYPE;
```

```
TYPE IfcStructuralCurveActivityTypeEnum = ENUMERATION OF  
(CONST,  
LINEAR,  
POLYGONAL,  
EQUIDISTANT,  
SINUS,  
PARABOLA,  
DISCRETE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcStructuralSurfaceActivityTypeEnum = ENUMERATION OF  
(CONST,  
BILINEAR,
```

```
DISCRETE,  
ISOCONTOUR,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcGlobalOrLocalEnum = ENUMERATION OF  
(GLOBAL_COORDS,  
LOCAL_COORDS);  
END_TYPE;
```

```
TYPE IfcStructuralActivityAssignmentSelect = SELECT  
(IfcElement,  
IfcStructuralItem);  
END_TYPE;
```

```
TYPE IfcModulusOfTranslationalSubgradeReactionSelect = SELECT  
(IfcBoolean,  
IfcModulusOfLinearSubgradeReactionMeasure);  
END_TYPE;
```

```
TYPE IfcModulusOfRotationalSubgradeReactionSelect = SELECT  
(IfcBoolean,  
IfcModulusOfRotationalSubgradeReactionMeasure);  
END_TYPE;
```

```
TYPE IfcModulusOfSubgradeReactionSelect = SELECT  
(IfcBoolean,  
IfcModulusOfSubgradeReactionMeasure);  
END_TYPE;
```

```
TYPE IfcWarpingStiffnessSelect = SELECT
  (IfcBoolean,
   IfcWarpingMomentMeasure);
END_TYPE;
```

```
TYPE IfcTranslationalStiffnessSelect = SELECT
  (IfcBoolean,
   IfcLinearStiffnessMeasure);
END_TYPE;
```

```
TYPE IfcRotationalStiffnessSelect = SELECT
  (IfcBoolean,
   IfcRotationalStiffnessMeasure);
END_TYPE;
```

```
TYPE IfcStructuralCurveMemberTypeEnum = ENUMERATION OF
  (RIGID_JOINED_MEMBER,
   PIN_JOINED_MEMBER,
   CABLE,
   TENSION_MEMBER,
   COMPRESSION_MEMBER,
   USERDEFINED,
   NOTDEFINED);
END_TYPE;
```

```
TYPE IfcStructuralSurfaceMemberTypeEnum = ENUMERATION OF
  (BENDING_ELEMENT,
   MEMBRANE_ELEMENT,
   SHELL,
```



```
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAlignmentTypeEnum = ENUMERATION OF  
(ABSOLUTE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcGridPlacementDirectionSelect = SELECT  
(IfcDirection,  
IfcVirtualGridIntersection);  
END_TYPE;
```

```
TYPE IfcProductSelect = SELECT  
(IfcProduct,  
IfcTypeProduct);  
END_TYPE;
```

```
TYPE IfcEventTypeEnum = ENUMERATION OF  
(STARTEVENT,  
ENDEVENT,  
INTERMEDIATEEVENT,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcEventTriggerTypeEnum = ENUMERATION OF  
(EVENTRULE,
```

```
EVENTMESSAGE,  
EVENTTIME,  
EVENTCOMPLEX,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcProcedureTypeEnum = ENUMERATION OF
```

```
(ADVICE_CAUTION,  
ADVICE_NOTE,  
ADVICE_WARNING,  
CALIBRATION,  
DIAGNOSTIC,  
SHUTDOWN,  
STARTUP,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTaskTypeEnum = ENUMERATION OF
```

```
(ATTENDANCE,  
CONSTRUCTION,  
DEMOLITION,  
DISMANTLE,  
DISPOSAL,  
INSTALLATION,  
LOGISTIC,  
MAINTENANCE,  
MOVE,  
OPERATION,
```

REMOVAL,
RENOVATION,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcProcessSelect = SELECT
(IfcProcess,
IfcTypeProcess);
END_TYPE;

TYPE IfcSequenceEnum = ENUMERATION OF
(START_START,
START_FINISH,
FINISH_START,
FINISH_FINISH,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcConstructionEquipmentResourceTypeEnum = ENUMERATION OF
(DEMOLISHING,
EARTHMOVING,
ERECTING,
HEATING,
LIGHTING,
PAVING,
PUMPING,
TRANSPORTING,
USERDEFINED,

NOTDEFINED);
END_TYPE;

TYPE IfcConstructionMaterialResourceTypeEnum = ENUMERATION OF

(AGGREGATES,
CONCRETE,
DRYWALL,
FUEL,
GYPSUM,
MASONRY,
METAL,
PLASTIC,
WOOD,
ASPHALT_K,
GRANITE_K,
PLYWOOD_K,
RUBBLE_K,
ASCON_K,
BROKENSTONE_K,
STOPPERSTONE_K,
LATEXMODIFIEDCONCRETE_K,
NOTDEFINED,
USERDEFINED);

END_TYPE;

TYPE IfcEarthworkMaterialResourceTypeEnum_K = ENUMERATION OF

(GRAVEL_K,
SAND_K,
SILT_K,
CLAY_K,

```
ORGANIC_CLAY_K,  
PEAT_K,  
REAPING_ROCK_K,  
BLASTING_ROCK_SOFT_K,  
BLASTING_ROCK_NORMAL_K,  
BLASTING_ROCK_HARD_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcConstructionProductResourceTypeEnum = ENUMERATION OF  
(ASSEMBLY,  
FORMWORK,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCrewResourceTypeEnum = ENUMERATION OF  
(OFFICE,  
SITE,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcLaborResourceTypeEnum = ENUMERATION OF  
(ADMINISTRATION,  
CARPENTRY,  
CLEANING,  
CONCRETE,  
DRYWALL,
```

ELECTRIC,
FINISHING,
FLOORING,
GENERAL,
HVAC,
LANDSCAPING,
MASONRY,
PAINTING,
PAVING,
PLUMBING,
ROOFING,
SITEGRADING,
STEELWORK,
SURVEYING,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcSubContractResourceTypeEnum = ENUMERATION OF
(PURCHASE,
WORK,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcResourceSelect = SELECT
(IfcResource,
IfcTypeResource);
END_TYPE;

```
TYPE IfcPropertySetDefinitionSelect = SELECT
  (IfcPropertySetDefinition,
   IfcPropertySetDefinitionSet);
END_TYPE;
```

```
TYPE IfcComplexPropertyTemplateTypeEnum = ENUMERATION OF
  (P_COMPLEX,
   Q_COMPLEX);
END_TYPE;
```

```
TYPE IfcSimplePropertyTemplateTypeEnum = ENUMERATION OF
  (P_SINGLEVALUE,
   P_ENUMERATEDVALUE,
   P_BOUNDEDVALUE,
   P_LISTVALUE,
   P_TABLEVALUE,
   P_REFERENCEVALUE,
   Q_LENGTH,
   Q_AREA,
   Q_VOLUME,
   Q_COUNT,
   Q_WEIGHT,
   Q_TIME);
END_TYPE;
```

```
TYPE IfcObjectReferenceSelect = SELECT
  (IfcAddress,
   IfcAppliedValue,
   IfcExternalReference,
   IfcMaterialDefinition,
```

```
    IfcOrganization,  
    IfcPerson,  
    IfcPersonAndOrganization,  
    IfcTable,  
    IfcTimeSeries);  
END_TYPE;
```

```
TYPE IfcClassificationReferenceSelect = SELECT  
    (IfcClassification,  
     IfcClassificationReference);  
END_TYPE;
```

```
TYPE IfcDocumentConfidentialityEnum = ENUMERATION OF  
    (PUBLIC,  
     RESTRICTED,  
     CONFIDENTIAL,  
     PERSONAL,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDocumentStatusEnum = ENUMERATION OF  
    (DRAFT,  
     FINALDRAFT,  
     FINAL,  
     REVISION,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDocumentSelect = SELECT
```



```
( IfcDocumentInformation,  
  IfcDocumentReference);  
END_TYPE;
```

```
TYPE IfcLibrarySelect = SELECT  
  ( IfcLibraryInformation,  
    IfcLibraryReference);  
END_TYPE;
```

```
TYPE IfcClassificationSelect = SELECT  
  ( IfcClassification,  
    IfcClassificationReference);  
END_TYPE;
```

```
TYPE IfcTimeSeriesDataTypeEnum = ENUMERATION OF  
  (CONTINUOUS,  
   DISCRETE,  
   DISCRETEBINARY,  
   PIECEWISEBINARY,  
   PIECEWISECONSTANT,  
   PIECEWISECONTINUOUS,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCurveInterpolationEnum = ENUMERATION OF  
  (LINEAR,  
   LOG_LINEAR,  
   LOG_LOG,  
   NOTDEFINED);  
END_TYPE;
```

TYPE IfcResourceObjectSelect = SELECT

(IfcActorRole,
IfcAppliedValue,
IfcApproval,
IfcConstraint,
IfcContextDependentUnit,
IfcConversionBasedUnit,
IfcExternalInformation,
IfcExternalReference,
IfcMaterialDefinition,
IfcOrganization,
IfcPerson,
IfcPersonAndOrganization,
IfcPhysicalQuantity,
IfcProfileDef,
IfcPropertyAbstraction,
IfcTimeSeries);

END_TYPE;

TYPE IfcBenchmarkEnum = ENUMERATION OF

(GREATERTHAN,
GREATERTHANOREQUALTO,
LESSTHAN,
LESSTHANOREQUALTO,
EQUALTO,
NOTEQUALTO,
INCLUDES,
NOTINCLUDES,
INCLUDEDIN,

```
NOT INCLUDED IN);  
END_TYPE;
```

```
TYPE IfcMetricValueSelect = SELECT  
  (IfcAppliedValue,  
   IfcMeasureWithUnit,  
   IfcReference,  
   IfcTable,  
   IfcTimeSeries,  
   IfcValue);  
END_TYPE;
```

```
TYPE IfcLogicalOperatorEnum = ENUMERATION OF  
  (LOGICALAND,  
   LOGICALOR,  
   LOGICALXOR,  
   LOGICALNOTAND,  
   LOGICALNOTOR);  
END_TYPE;
```

```
TYPE IfcObjectiveEnum = ENUMERATION OF  
  (CODECOMPLIANCE,  
   CODEWAIVER,  
   DESIGNINTENT,  
   EXTERNAL,  
   HEALTHANDSAFETY,  
   MERGECONFLICT,  
   MODELVIEW,  
   PARAMETER,  
   REQUIREMENT,
```

SPECIFICATION,
TRIGGERCONDITION,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcConstraintEnum = ENUMERATION OF
(HARD,
SOFT,
ADVISORY,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcReinforcingBarSurfaceEnum = ENUMERATION OF
(PLAIN,
TEXTURED);
END_TYPE;

TYPE IfcSectionTypeEnum = ENUMERATION OF
(UNIFORM,
TAPERED);
END_TYPE;

TYPE IfcReinforcingBarRoleEnum = ENUMERATION OF
(MAIN,
SHEAR,
LIGATURE,
STUD,
PUNCHING,

```
EDGE,  
RING,  
ANCHORING,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcStateEnum = ENUMERATION OF  
(READWRITE,  
READONLY,  
LOCKED,  
READWRITELOCKED,  
READONLYLOCKED);  
END_TYPE;
```

```
TYPE IfcPropertySetTemplateTypeEnum = ENUMERATION OF  
(PSET_TYPEDRIVENONLY,  
PSET_TYPEDRIVENOVERRIDE,  
PSET_OCCURRENCEDRIVEN,  
PSET_PERFORMANCEDRIVEN,  
QTO_TYPEDRIVENONLY,  
QTO_TYPEDRIVENOVERRIDE,  
QTO_OCCURRENCEDRIVEN,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorStyleOperationEnum = ENUMERATION OF  
(SINGLE_SWING_LEFT,  
SINGLE_SWING_RIGHT,  
DOUBLE_DOOR_SINGLE_SWING,
```

```
DOUBLE_DOOR_SINGLE_SWING_OPPOSITE_LEFT,  
DOUBLE_DOOR_SINGLE_SWING_OPPOSITE_RIGHT,  
DOUBLE_SWING_LEFT,  
DOUBLE_SWING_RIGHT,  
DOUBLE_DOOR_DOUBLE_SWING,  
SLIDING_TO_LEFT,  
SLIDING_TO_RIGHT,  
DOUBLE_DOOR_SLIDING,  
FOLDING_TO_LEFT,  
FOLDING_TO_RIGHT,  
DOUBLE_DOOR_FOLDING,  
REVOLVING,  
ROLLINGUP,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorStyleConstructionEnum = ENUMERATION OF  
(ALUMINIUM,  
HIGH_GRADE_STEEL,  
STEEL,  
WOOD,  
ALUMINIUM_WOOD,  
ALUMINIUM_PLASTIC,  
PLASTIC,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcRoadSignEquipmentTypeEnum_K = ENUMERATION OF
```

```
(DELINEATOR_K,  
TRAFFICSIGN_K,  
REFLECTINGMIRROR_K,  
TRAFFICIGNAL_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPavementAdditionTypeEnum_K = ENUMERATION OF  
(PEDESTRIANCROSSWALK_K,  
ROADMARKER_K,  
SPEEDHUMP_K,  
ANTISLIDING_K,  
RUMBLESTRIP_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcGuardTypeEnum_K = ENUMERATION OF  
(CRASH_CUSHION_K,  
SOUNDPROOF_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcEarthworkElementTypeEnum_K = ENUMERATION OF  
(CUTTING_K,  
FILLING_K,  
CUTTINGANDFILLING_K,  
USERDEFINED,
```

NOTDEFINED);
END_TYPE;

TYPE IfcRoadShoulderTypeEnum_K = ENUMERATION OF
(FULLWIDTH_K,
HALFWIDTH_K,
NARROWWIDTH_K,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcRoadBodyTypeEnum_K = ENUMERATION OF
(USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcRoadMedianStripTypeEnum = ENUMERATION OF
(GUARDFENCEOFCONCRETE_K,
GUARDRAIL_K,
GREENAREA_K,
CURBOFCONCRETE_K,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcCurbTypeEnum_K = ENUMERATION OF
(MOUNTABLE_CURB_K,
BARRIER1_K,
BARRIER2_K,
USERDEFINED,

NOTDEFINED);
END_TYPE;

TYPE IfcPavementTypeEnum_K = ENUMERATION OF

(SURFACE_K,
INTERMEDIATECOURSE_K,
SUBBASE_K,
BASECOURSE_K,
LEANCONCRETE_K,
ANTIFREEZINGLAYER_K,
BRIDGEDECKSSURFACING_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcRetainingWallTypeEnum_K = ENUMERATION OF

(GRAVITY_TYPE_K,
SEMIGRAVITY_TYPE_K,
NON_STANDING_K,
CANTILEVER_TYPE_K,
L_SHAPED_K,
REVERSED_T_SHAPED_K,
COUNTERFORT_K,
MASONARY_K,
REINFORCED_EARTH_K,
BLOCK_TYPE_K,
PANEL_TYPE_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcCulvertTypeEnum_K = ENUMERATION OF

(ONEWAY_WATERWAY_K,
TWOWAY_WATERWAY_K,
THREEWAY_WATERWAY_K,
ONEWAY_PASSAGEWAY_K,
TWOWAY_PASSAGEWAY_K,
ONEWAY_SKEWED_WATERWAY_K,
TWOWAY_SKEWED_WATERWAY_K,
THREEWAY_SKEWED_WATERWAY_K,
ONEWAY_SKEWED_PASSAGEWAY_K,
TWOWAY_SKEWED_PASSAGEWAY_K,
STONEFILLED_CULVERT_TYPE1_K,
STONEFILLED_CULVERT_TYPE2_K,
STONEFILLED_CULVERT_TYPE3_K,
STONEFILLED_CULVERT_TYPE4_K,
STONEFILLED_CULVERT_TYPE5_K,
COMMONDUCT_K,
NOTDEFINED,
USERDEFINED);

END_TYPE;

TYPE IfcCaissonTypeEnum_K = ENUMERATION OF

(OPEN_K,
BOX_K,
PNEUMATIC_K,
FLOATING_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

```
TYPE IfcBridgeDeckTypeEnum_K = ENUMERATION OF
  (APPROACH_SLAB,
   USER_DEFINED,
   NOT_DEFINED);
END_TYPE;
```

```
TYPE IfcBridgeTowerTypeEnum_K = ENUMERATION OF
  (A_SHAPED_K,
   H_SHAPED_K,
   I_SHAPED_K,
   DIAMOND_SHAPED_K,
   NOTDEFINED,
   USERDEFINED);
END_TYPE;
```

```
TYPE IfcBridgeCableTypeEnum_K = ENUMERATION OF
  (SUSPENDER_K,
   SUSPENSIONCABLE_K,
   TENSIONCABLE_K,
   USERDEFINED,
   NOTDEFINED);
END_TYPE;
```

```
TYPE IfcBridgePierTypeEnum_K = ENUMERATION OF
  (GRAVITY_TYPE_K,
   WALL_TYPE_K,
   RAHMEN_PIER_TYPE_K,
   T_SHAPED_TYPE_K,
   RAHMEN_ABUT_TYPE_K,
```

```
ARCH_TYPE_K,  
V_SHAPED_TYPE_K,  
SEMI_GRAVITY_TYPE_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBridgeAbutmentTypeEnum_K = ENUMERATION OF  
(GRAVITY_TYPE_K,  
SEMI_GRAVITY_TYPE_K,  
REVERSED_T_SHAPED_TYPE_K,  
COUNTERFORT_TYPE_K,  
RAHMEN_TYPE_K,  
RAHMEN_ABUT_TYPE_K,  
BOX_TYPE_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBridgeSpanTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBridgeSegmentTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBridgeCopingTypeEnum_K = ENUMERATION OF
```

```
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBridgeGirderTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTunnelLiningTypeEnum_K = ENUMERATION OF  
(GROUNDREINFORCING_BODY_K,  
SHORTCRETE_BODY_K,  
LINING_BODY_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTunnelLiningSegmentTypeEnum_K = ENUMERATION OF  
(STEEL_SEGMENT_K,  
CONCRETE_SEGMENT_K,  
CAST_IRON_SEGMENT_K,  
COMBINED_SEGMENT_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCivilElementProxyTypeEnum_K = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

TYPE IfcElementAssemblyTypeEnum = ENUMERATION OF

(ACCESSORY_ASSEMBLY,
ARCH,
BEAM_GRID,
BRACED_FRAME,
GIRDER,
REINFORCEMENT_UNIT,
RIGID_FRAME,
SLAB_FIELD,
TRUSS,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcBuildingElementPartTypeEnum = ENUMERATION OF

(INSULATION,
PRECASTPANEL,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcDiscreteAccessoryTypeEnum = ENUMERATION OF

(ANCHORPLATE,
BRACKET,
SHOE,
STEELGRATING_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcFastenerTypeEnum = ENUMERATION OF

(GLUE,
MORTAR,
WELD,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcMechanicalFastenerTypeEnum = ENUMERATION OF

(ANCHORBOLT,
BOLT,
DOWEL,
NAIL,
NAILPLATE,
RIVET,
SCREW,
SHEARCONNECTOR,
STAPLE,
STUDSHEARCONNECTOR,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcReinforcingBarTypeEnum = ENUMERATION OF

(ANCHORING,
EDGE,
LIGATURE,
MAIN,
PUNCHING,

```
RING,  
SHEAR,  
STUD,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBendingParameterSelect = SELECT  
  (IfcLengthMeasure,  
   IfcPlaneAngleMeasure);  
END_TYPE;
```

```
TYPE IfcReinforcingMeshTypeEnum = ENUMERATION OF  
  (USERDEFINED,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTendonAnchorTypeEnum = ENUMERATION OF  
  (COUPLER,  
   FIXED_END,  
   TENSIONING_END,  
   USERDEFINED,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTendonTypeEnum = ENUMERATION OF  
  (BAR,  
   COATED,  
   STRAND,  
   WIRE,
```



```
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcGroundReinforcingElementyTypeEnum_K = ENUMERATION OF
```

```
(EARTH_ANCHOR_K,  
ROCK_BOLT_K,  
ROCK_ANCHOR_K,  
SHOTCRETE_K,  
STEEL_RIB_K,  
STEEL_FIBER_K,  
SYNTHETIC_FIBER_K,  
PILE_K,  
WIREROPE_K,  
BLOCK_K,  
CONCRETE_K,  
STONE_K,  
TEXTILE_K,  
NET_K,  
PLANT_K,  
COUNTERWEIGHT_FILL_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcVibrationIsolatorTypeEnum = ENUMERATION OF
```

```
(COMPRESSION,  
SPRING,  
USERDEFINED,  
NOTDEFINED);
```

END_TYPE;

TYPE IfcWaterProofElementTypeEnum_K = ENUMERATION OF

(WATERPROOF_SHEET_K,
FLUID_APPLIED_K,
ASPHALT_K,
SEALING_K,
CAULKING_K,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcRoadElementPartTypeEnum_K = ENUMERATION OF

(STABILIZATIONFILTER_K,
SPACER_K,
RAILJOINT_K,
EXPANSIONJOINT_K,
TRANSVERSECONTRACTIONJOINT_K,
LONGITUDINALJOINT_K,
CONSTRUCTIONJOINT_K,
STYROFOAM_K,
WIREMESH_K,
PRIMECOAT_K,
TACKCOAT_K,
GEOTEXTILE_K,
DOWERBAR_K,
TIEBAR_K,
SLIPBAR_K,
BEDJOINT_K,
CUTTINGJOINT_K,

```
FOOTPATHBOUNDARYSTONE_K,  
ROADBOUNDARYSTONE_K,  
INSPECTIONLADDER_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcBridgeElementPartTypeEnum_K = ENUMERATION OF
```

```
(FLANGE_K,  
OVERHANG_K,  
FLOORING_K,  
VERTICAL_STIFFENER_K,  
LONGITUDINAL_STIFFENER_K,  
WEB_K,  
RIB_K,  
PLATE_K,  
STRAND_K,  
BRACING_K,  
GIRDERLINKAGE_K,  
ANCHORAGE_K,  
PIER_SCOUR_PROTECTION_K,  
SHIP_IMPACT_PROTECTION_K,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcTunnelElementPartTypeEnum_K = ENUMERATION OF
```

```
(SEGMENT_JOINT_K,  
RING_JOINT_K,  
USERDEFINED,
```

```
    NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcAssemblyPlaceEnum = ENUMERATION OF  
    (SITE,  
     FACTORY,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcFurnitureTypeEnum = ENUMERATION OF  
    (CHAIR,  
     TABLE,  
     DESK,  
     BED,  
     FILECABINET,  
     SHELF,  
     SOFA,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcSystemFurnitureElementTypeEnum = ENUMERATION OF  
    (PANEL,  
     WORKSURFACE,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcGeographicElementTypeEnum = ENUMERATION OF  
    (TERRAIN,
```

USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcTransportElementTypeEnum = ENUMERATION OF
(ELEVATOR,
ESCALATOR,
MOVINGWALKWAY,
CRANEWAY,
LIFTINGGEAR,
USERDEFINED,
NOTDEFINED);
END_TYPE;

TYPE IfcWindowStyleConstructionEnum = ENUMERATION OF
(ALUMINIUM,
HIGH_GRADE_STEEL,
STEEL,
WOOD,
ALUMINIUM_WOOD,
PLASTIC,
OTHER_CONSTRUCTION,
NOTDEFINED);
END_TYPE;

TYPE IfcWindowStyleOperationEnum = ENUMERATION OF
(SINGLE_PANEL,
DOUBLE_PANEL_VERTICAL,
DOUBLE_PANEL_HORIZONTAL,
TRIPLE_PANEL_VERTICAL,

```
TRIPLE_PANEL_BOTTOM,  
TRIPLE_PANEL_TOP,  
TRIPLE_PANEL_LEFT,  
TRIPLE_PANEL_RIGHT,  
TRIPLE_PANEL_HORIZONTAL,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcProjectionElementTypeEnum = ENUMERATION OF  
(USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcOpeningElementTypeEnum = ENUMERATION OF  
(OPENING,  
RECESS,  
USERDEFINED,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcVoidingFeatureTypeEnum = ENUMERATION OF  
(CUTOUT,  
NOTCH,  
HOLE,  
MITER,  
CHAMFER,  
EDGE,  
USERDEFINED,  
NOTDEFINED);
```

END_TYPE;

TYPE IfcSurfaceFeatureTypeEnum = ENUMERATION OF

(MARK,
TAG,
TREATMENT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcAnalysisModelTypeEnum = ENUMERATION OF

(IN_PLANE_LOADING_2D,
OUT_PLANE_LOADING_2D,
LOADING_3D,
USERDEFINED,
NOTDEFINED);

END_TYPE;

TYPE IfcOccupantTypeEnum = ENUMERATION OF

(ASSIGNEE,
ASSIGNOR,
LESSEE,
LESSOR,
LETTINGAGENT,
OWNER,
TENANT,
USERDEFINED,
NOTDEFINED);

END_TYPE;

```
TYPE IfcDefinitionSelect = SELECT
  (IfcObjectDefinition,
   IfcPropertyDefinition);
END_TYPE;
```

```
TYPE IfcMaterialSelect = SELECT
  (IfcMaterialDefinition,
   IfcMaterialList,
   IfcMaterialUsageDefinition);
END_TYPE;
```

```
TYPE IfcDirectionSenseEnum = ENUMERATION OF
  (POSITIVE,
   NEGATIVE);
END_TYPE;
```

```
TYPE IfcCurveFontOrScaledCurveFontSelect = SELECT
  (IfcCurveStyleFontAndScaling,
   IfcCurveStyleFontSelect);
END_TYPE;
```

```
TYPE IfcFillStyleSelect = SELECT
  (IfcColour,
   IfcExternallyDefinedHatchStyle,
   IfcFillAreaStyleHatching,
   IfcFillAreaStyleTiles);
END_TYPE;
```

```
TYPE IfcHatchLineDistanceSelect = SELECT
  (IfcPositiveLengthMeasure,
```



```
    IfcVector);  
END_TYPE;
```

```
TYPE IfcStyleAssignmentSelect = SELECT  
    (IfcPresentationStyle,  
     IfcPresentationStyleAssignment);  
END_TYPE;
```

```
TYPE IfcPresentationStyleSelect = SELECT  
    (IfcCurveStyle,  
     IfcFillAreaStyle,  
     IfcNullStyle,  
     IfcSurfaceStyle,  
     IfcTextStyle);  
END_TYPE;
```

```
TYPE IfcNullStyle = ENUMERATION OF  
    (NULL);  
END_TYPE;
```

```
TYPE IfcSurfaceSide = ENUMERATION OF  
    (POSITIVE,  
     NEGATIVE,  
     BOTH);  
END_TYPE;
```

```
TYPE IfcSurfaceStyleElementSelect = SELECT  
    (IfcExternallyDefinedSurfaceStyle,  
     IfcSurfaceStyleLighting,  
     IfcSurfaceStyleRefraction,
```

```
    IfcSurfaceStyleShading,  
    IfcSurfaceStyleWithTextures);  
END_TYPE;
```

```
TYPE IfcTextFontSelect = SELECT  
    (IfcExternallyDefinedTextFont,  
    IfcPreDefinedTextFont);  
END_TYPE;
```

```
TYPE IfcLayeredItem = SELECT  
    (IfcRepresentation,  
    IfcRepresentationItem);  
END_TYPE;
```

```
TYPE IfcGeometricSetSelect = SELECT  
    (IfcCurve,  
    IfcPoint,  
    IfcSurface);  
END_TYPE;
```

```
TYPE IfcLightEmissionSourceEnum = ENUMERATION OF  
    (COMPACTFLUORESCENT,  
    FLUORESCENT,  
    HIGHPRESSUREMERCURY,  
    HIGHPRESSURESODIUM,  
    LIGHTEMITTINGDIODE,  
    LOWPRESSURESODIUM,  
    LOWVOLTAGEHALOGEN,  
    MAINVOLTAGEHALOGEN,  
    METALHALIDE,
```

```
TUNGSTENFILAMENT,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcLightDistributionDataSourceSelect = SELECT  
  (IfcExternalReference,  
   IfcLightIntensityDistribution);  
END_TYPE;
```

```
TYPE IfcLightDistributionCurveEnum = ENUMERATION OF  
  (TYPE_A,  
   TYPE_B,  
   TYPE_C,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcShell = SELECT  
  (IfcClosedShell,  
   IfcOpenShell);  
END_TYPE;
```

```
TYPE IfcTextPath = ENUMERATION OF  
  (LEFT,  
   RIGHT,  
   UP,  
   DOWN);  
END_TYPE;
```

```
TYPE IfcCoordinateReferenceSystemSelect = SELECT  
  (IfcCoordinateReferenceSystem,
```

```
    IfcGeometricRepresentationContext);  
END_TYPE;
```

```
TYPE IfcChangeActionEnum = ENUMERATION OF  
    (NOCHANGE,  
     MODIFIED,  
     ADDED,  
     DELETED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorPanelOperationEnum = ENUMERATION OF  
    (SWINGING,  
     DOUBLE_ACTING,  
     SLIDING,  
     FOLDING,  
     REVOLVING,  
     ROLLINGUP,  
     FIXEDPANEL,  
     USERDEFINED,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcDoorPanelPositionEnum = ENUMERATION OF  
    (LEFT,  
     MIDDLE,  
     RIGHT,  
     NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcPermeableCoveringOperationEnum = ENUMERATION OF  
  (GRILL,  
   LOUVER,  
   SCREEN,  
   USERDEFINED,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcWindowPanelPositionEnum = ENUMERATION OF  
  (LEFT,  
   MIDDLE,  
   RIGHT,  
   BOTTOM,  
   TOP,  
   NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcWindowPanelOperationEnum = ENUMERATION OF  
  (SIDEHUNGRIGHTHAND,  
   SIDEHUNGLEFTHAND,  
   TILTANDTURNRIGHTHAND,  
   TILTANDTURNLEFTHAND,  
   TOPHUNG,  
   BOTTOMHUNG,  
   PIVOTHORIZONTAL,  
   PIVOTVERTICAL,  
   SLIDINGHORIZONTAL,  
   SLIDINGVERTICAL,  
   REMOVABLECASEMENT,  
   FIXEDCASEMENT,
```

```
OTHEROPERATION,  
NOTDEFINED);  
END_TYPE;
```

```
TYPE IfcCurveOnSurface = SELECT  
  (IfcCompositeCurveOnSurface,  
   IfcPcurve);  
END_TYPE;
```

```
TYPE IfcVectorOrDirection = SELECT  
  (IfcDirection,  
   IfcVector);  
END_TYPE;
```

```
ENTITY IfcPropertySetDefinition  
  ABSTRACT SUPERTYPE OF (ONEOF(IfcPreDefinedPropertySet, IfcPropertySet, IfcQuantitySet))  
  SUBTYPE OF(IfcPropertyDefinition);  
  INVERSE  
    DefinesType      : SET OF IfcTypeObject FOR HasPropertySets;  
    IsDefinedBy      : SET OF IfcRelDefinesByTemplate FOR RelatedPropertySets;  
    DefinesOccurrence : SET OF IfcRelDefinesByProperties FOR RelatingPropertyDefinition;  
END_ENTITY;
```

```
ENTITY IfcPropertyDefinition  
  ABSTRACT SUPERTYPE OF (ONEOF(IfcPropertySetDefinition, IfcPropertyTemplateDefinition))  
  SUBTYPE OF(IfcRoot);  
  INVERSE  
    HasContext      : SET [0:1] OF IfcRelDeclares FOR RelatedDefinitions;  
    HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;  
END_ENTITY;
```

ENTITY IfcRoot

ABSTRACT SUPERTYPE OF (ONEOF(IfcObjectDefinition, IfcPropertyDefinition, IfcRelationship));

GlobalId : IfcGloballyUniqueId;

OwnerHistory : OPTIONAL IfcOwnerHistory;

Name : OPTIONAL IfcLabel;

Description : OPTIONAL IfcText;

UNIQUE

UR1 : GlobalId;

END_ENTITY;

ENTITY IfcObjectDefinition

ABSTRACT SUPERTYPE OF (ONEOF(IfcContext, IfcObject, IfcTypeObject))

SUBTYPE OF(IfcRoot);

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;

END_ENTITY;

ENTITY IfcContext

ABSTRACT SUPERTYPE OF (ONEOF(IfcProject, IfcProjectLibrary))

SUBTYPE OF(IfcObjectDefinition);

ObjectType : OPTIONAL IfcLabel;

LongName : OPTIONAL IfcLabel;

Phase : OPTIONAL IfcLabel;

```

RepresentationContexts : OPTIONAL SET [1:?] OF IfcRepresentationContext;
UnitsInContext          : OPTIONAL IfcUnitAssignment;

INVERSE

IsDefinedBy             : SET OF IfcRelDefinesByProperties FOR RelatedObjects;
Declares                 : SET OF IfcRelDeclares FOR RelatingContext;

END_ENTITY;

ENTITY IfcProject
  SUBTYPE OF(IfcContext);
  WHERE
    HASNAME              : EXISTS(SELFIfcRoot.Name);
    CORRECTCONTEXT       : NOT(EXISTS(SELFIfcContext.RepresentationContexts)) OR
      (SIZEOF(QUERY(Temp <* SELFIfcContext.RepresentationContexts |
        'IFC4.IFCGEOMETRICREPRESENTATIONSUBCONTEXT' IN TYPEOF(Temp)
      )) = 0);
    NODECOMPOSITION      : SIZEOF(SELFIfcObjectDefinition.Decomposes) = 0;
END_ENTITY;

ENTITY IfcProjectLibrary
  SUBTYPE OF(IfcContext);
END_ENTITY;

ENTITY IfcRepresentationContext
  ABSTRACT SUPERTYPE;
  ContextIdentifier      : OPTIONAL IfcLabel;
  ContextType            : OPTIONAL IfcLabel;
  INVERSE
    RepresentationsInContext : SET OF IfcRepresentation FOR ContextOfItems;
END_ENTITY;

```


ENTITY IfcGeometricRepresentationContext

SUBTYPE OF(IfcRepresentationContext);

CoordinateSpaceDimension : IfcDimensionCount;

Precision : OPTIONAL IfcReal;

WorldCoordinateSystem : IfcAxis2Placement;

TrueNorth : OPTIONAL IfcDirection;

INVERSE

HasSubContexts : SET OF IfcGeometricRepresentationSubContext FOR ParentContext;

HasCoordinateOperation : SET [0:1] OF IfcCoordinateOperation FOR SourceCRS;

WHERE

NORTH2D : NOT(EXISTS(TrueNorth)) OR (HINDEX(TrueNorth.DirectionRatios) = 2);

END_ENTITY;

ENTITY IfcGeometricRepresentationSubContext

SUBTYPE OF(IfcGeometricRepresentationContext);

ParentContext :
IfcGeometricRepresentationContext;

TargetScale : OPTIONAL
IfcPositiveRatioMeasure;

TargetView :
IfcGeometricProjectionEnum;

UserDefinedTargetView : OPTIONAL IfcLabel;

DERIVE

SELFIfcGeometricRepresentationContext.WorldCoordinateSystem : IfcAxis2Placement :=
ParentContext.WorldCoordinateSystem;

SELFIfcGeometricRepresentationContext.CoordinateSpaceDimension : IfcDimensionCount :=
ParentContext.CoordinateSpaceDimension;

SELFIfcGeometricRepresentationContext.TrueNorth : IfcDirection :=
NVL(ParentContext.TrueNorth,
IfcConvertDirectionInto2D(SELFIfcGeometricRepresentationContext.WorldCoordinateSystem.P[2]));

SELFIfcGeometricRepresentationContext.Precision : IfcReal :=
NVL(ParentContext.Precision, 1.E-5);

```

WHERE
    PARENTNOSUB          : NOT('IFC4.IFCGEOMETRICREPRESENTATIONSUBCONTEXT' IN
    TYPEOF(ParentContext));
    USERTARGETPROVIDED : (TargetView <> IfcGeometricProjectionEnum.USERDEFINED) OR
                        ((TargetView = IfcGeometricProjectionEnum.USERDEFINED) AND
    EXISTS(UserDefinedTargetView));
    NOCOORDOPERATION   : SIZEOF(SELFWIfcGeometricRepresentationContext.HasCoordinateOperation)
    = 0;
END_ENTITY;

```

```

ENTITY IfcAxis2Placement2D
    SUBTYPE OF(IfcPlacement);
    RefDirection : OPTIONAL IfcDirection;
    DERIVE
    P          : LIST [2:2] OF IfcDirection := IfcBuild2Axes(RefDirection);
    WHERE
    REFDIRIS2D : (NOT (EXISTS (RefDirection))) OR (RefDirection.Dim = 2);
    LOCATIONIS2D : SELFWIfcPlacement.Location.Dim = 2;
END_ENTITY;

```

```

ENTITY IfcPlacement
    ABSTRACT SUPERTYPE OF (ONEOF(IfcAxis1Placement, IfcAxis2Placement2D, IfcAxis2Placement3D))
    SUBTYPE OF(IfcGeometricRepresentationItem);
    Location : IfcCartesianPoint;
    DERIVE
    Dim      : IfcDimensionCount := Location.Dim;
END_ENTITY;

```

```

ENTITY IfcGeometricRepresentationItem
    ABSTRACT SUPERTYPE OF (ONEOF(IfcAnnotationFillArea, IfcBooleanResult, IfcBoundingBox,
    IfcCartesianPointList, IfcCartesianTransformationOperator, IfcCompositeCurveSegment,

```

IfcCsgPrimitive3D, IfcCurve, IfcDirection, IfcFaceBasedSurfaceModel, IfcFillAreaStyleHatching, IfcFillAreaStyleTiles, IfcGeometricSet, IfcHalfSpaceSolid, IfcLightSource, IfcPlacement, IfcPlanarExtent, IfcPoint, IfcSectionedSpine, IfcShellBasedSurfaceModel, IfcSolidModel, IfcSurface, IfcTessellatedItem, IfcTextLiteral, IfcVector))

SUBTYPE OF(IfcRepresentationItem);

END_ENTITY;

ENTITY IfcRepresentationItem

ABSTRACT SUPERTYPE OF (ONEOF(IfcGeometricRepresentationItem, IfcMappedItem, IfcStyledItem, IfcTopologicalRepresentationItem));

INVERSE

LayerAssignment : SET [0:1] OF IfcPresentationLayerAssignment FOR AssignedItems;

StyledByItem : SET [0:1] OF IfcStyledItem FOR Item;

END_ENTITY;

ENTITY IfcMappedItem

SUBTYPE OF(IfcRepresentationItem);

MappingSource : IfcRepresentationMap;

MappingTarget : IfcCartesianTransformationOperator;

END_ENTITY;

ENTITY IfcRepresentationMap;

MappingOrigin : IfcAxis2Placement;

MappedRepresentation : IfcRepresentation;

INVERSE

HasShapeAspects : SET OF IfcShapeAspect FOR PartOfProductDefinitionShape;

MapUsage : SET OF IfcMappedItem FOR MappingSource;

WHERE

APPLICABLEMAPPEDREPR : 'IFC4.IFCSHAPEMODEL' IN TYPEOF(MappedRepresentation);

END_ENTITY;

ENTITY IfcRepresentation

ABSTRACT SUPERTYPE OF (ONEOF(IfcShapeModel, IfcStyleModel));

ContextOfItems : IfcRepresentationContext;

RepresentationIdentifier : OPTIONAL IfcLabel;

RepresentationType : OPTIONAL IfcLabel;

Items : SET [1:?] OF IfcRepresentationItem;

INVERSE

RepresentationMap : SET [0:1] OF IfcRepresentationMap FOR MappedRepresentation;

LayerAssignments : SET OF IfcPresentationLayerAssignment FOR AssignedItems;

OfProductRepresentation : SET OF IfcProductRepresentation FOR Representations;

END_ENTITY;

ENTITY IfcShapeModel

ABSTRACT SUPERTYPE OF (ONEOF(IfcShapeRepresentation, IfcTopologyRepresentation))

SUBTYPE OF(IfcRepresentation);

INVERSE

OfShapeAspect : SET [0:1] OF IfcShapeAspect FOR ShapeRepresentations;

WHERE

WR11 : (SIZEOF(SELFWIfcRepresentation.OfProductRepresentation) = 1) XOR

(SIZEOF(SELFWIfcRepresentation.RepresentationMap) = 1) XOR

(SIZEOF(OfShapeAspect) = 1);

END_ENTITY;

ENTITY IfcShapeRepresentation

SUBTYPE OF(IfcShapeModel);

WHERE

CORRECTCONTEXT : 'IFC4.IFCGEOMETRICREPRESENTATIONCONTEXT'

IN TYPEOF(SELFWIfcRepresentation.ContextOfItems);

NOTOLOGICALITEM : SIZEOF(QUERY(temp <* Items |

('IFC4.IFCTOPOLOGICALREPRESENTATIONITEM' IN TYPEOF(temp)))

```

        AND (NOT(SIZEOF(
        ['IFC4.IFCVERTEXPOINT',
        'IFC4.IFCEDGECURVE',
        'IFC4.IFCFACESURFACE'] * TYPEOF(temp)) = 1))
        )) = 0;

    HASREPRESENTATIONTYPE      : EXISTS(SELFWIfcRepresentation.RepresentationType);
    CORRECTITEMSFORTYPE      :
IfcShapeRepresentationTypes(SELFWIfcRepresentation.RepresentationType,
SELFWIfcRepresentation.Items);
    HASREPRESENTATIONIDENTIFIER : EXISTS(SELFWIfcRepresentation.RepresentationIdentifier);
END_ENTITY;

ENTITY IfcTopologyRepresentation
    SUBTYPE OF(IfcShapeModel);
    WHERE
        WR21 : SIZEOF(QUERY(temp <* SELFWIfcRepresentation.Items |
            NOT('IFC4.IFCTOPOLOGICALREPRESENTATIONITEM' IN TYPEOF(temp))
            )) = 0;
        WR22 : EXISTS(SELFWIfcRepresentation.RepresentationType);
        WR23      :      IfcTopologyRepresentationTypes(SELFWIfcRepresentation.RepresentationType,
SELFWIfcRepresentation.Items);
END_ENTITY;

ENTITY IfcShapeAspect;
    ShapeRepresentations      : LIST [1:?] OF IfcShapeModel;
    Name                      : OPTIONAL IfcLabel;
    Description                : OPTIONAL IfcText;
    ProductDefinitional       : IfcLogical;
    PartOfProductDefinitionShape : OPTIONAL IfcProductRepresentationSelect;
END_ENTITY;

```

```

ENTITY IfcProductDefinitionShape
  SUBTYPE OF(IfcProductRepresentation);
  INVERSE
    ShapeOfProduct : SET [1:?] OF IfcProduct FOR Representation;
    HasShapeAspects : SET OF IfcShapeAspect FOR PartOfProductDefinitionShape;
  WHERE
    ONLYSHAPEMODEL : SIZEOF(QUERY(temp <* Representations |
      (NOT('IFC4.IFCSHAPEMODEL' IN TYPEOF(temp)))
    )) = 0;
END_ENTITY;

```

```

ENTITY IfcProductRepresentation
  ABSTRACT SUPERTYPE OF (ONEOF(IfcMaterialDefinitionRepresentation, IfcProductDefinitionShape));
  Name : OPTIONAL IfcLabel;
  Description : OPTIONAL IfcText;
  Representations : LIST [1:?] OF IfcRepresentation;
END_ENTITY;

```

```

ENTITY IfcMaterialDefinitionRepresentation
  SUBTYPE OF(IfcProductRepresentation);
  RepresentedMaterial : IfcMaterial;
  WHERE
    ONLYSTYLEDREPRESENTATIONS : SIZEOF(QUERY(temp <* Representations |
      (NOT('IFC4.IFCSTYLEDREPRESENTATION' IN TYPEOF(temp)))
    )) = 0;
END_ENTITY;

```

```

ENTITY IfcMaterial
  SUBTYPE OF(IfcMaterialDefinition);
  Name : IfcLabel;

```

Description : OPTIONAL IfcText;
Category : OPTIONAL IfcLabel;

INVERSE

HasRepresentation : SET [0:1] OF IfcMaterialDefinitionRepresentation FOR RepresentedMaterial;
IsRelatedWith : SET OF IfcMaterialRelationship FOR RelatedMaterials;
RelatesTo : SET [0:1] OF IfcMaterialRelationship FOR RelatingMaterial;

END_ENTITY;

ENTITY IfcMaterialDefinition

ABSTRACT SUPERTYPE OF (ONEOF(IfcMaterial, IfcMaterialConstituent, IfcMaterialConstituentSet, IfcMaterialLayer, IfcMaterialLayerSet, IfcMaterialProfile, IfcMaterialProfileSet));

INVERSE

AssociatedTo : SET OF IfcRelAssociatesMaterial FOR RelatingMaterial;
HasExternalReferences : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;
HasProperties : SET OF IfcMaterialProperties FOR Material;

END_ENTITY;

ENTITY IfcMaterialConstituent

SUBTYPE OF(IfcMaterialDefinition);

Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
Material : IfcMaterial;
Fraction : OPTIONAL IfcNormalisedRatioMeasure;
Category : OPTIONAL IfcLabel;

INVERSE

ToMaterialConstituentSet : IfcMaterialConstituentSet FOR MaterialConstituents;

END_ENTITY;

ENTITY IfcMaterialConstituentSet

SUBTYPE OF(IfcMaterialDefinition);

Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
MaterialConstituents : OPTIONAL SET [1:?] OF IfcMaterialConstituent;
END_ENTITY;

ENTITY IfcMaterialLayer

SUBTYPE OF(IfcMaterialDefinition);

Material : OPTIONAL IfcMaterial;
LayerThickness : IfcNonNegativeLengthMeasure;
IsVentilated : OPTIONAL IfcLogical;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
Category : OPTIONAL IfcLabel;
Priority : OPTIONAL IfcInteger;

INVERSE

ToMaterialLayerSet : IfcMaterialLayerSet FOR MaterialLayers;

WHERE

NORMALIZEDPRIORITY : NOT(EXISTS(Priority)) OR {0 <= Priority <= 100};

END_ENTITY;

ENTITY IfcMaterialLayerWithOffsets

SUBTYPE OF(IfcMaterialLayer);

OffsetDirection : IfcLayerSetDirectionEnum;
OffsetValues : ARRAY [1:2] OF IfcLengthMeasure;

END_ENTITY;

ENTITY IfcMaterialLayerSet

SUBTYPE OF(IfcMaterialDefinition);

MaterialLayers : LIST [1:?] OF IfcMaterialLayer;
LayerSetName : OPTIONAL IfcLabel;


```

    Description    : OPTIONAL IfcText;
DERIVE
    TotalThickness : IfcLengthMeasure := IfcMlsTotalThickness(SELF);
END_ENTITY;

```

```

ENTITY IfcMaterialProfile

```

```

    SUBTYPE OF(IfcMaterialDefinition);

```

```

    Name           : OPTIONAL IfcLabel;
    Description    : OPTIONAL IfcText;
    Material       : OPTIONAL IfcMaterial;
    Profile        : IfcProfileDef;
    Priority       : OPTIONAL IfcInteger;
    Category       : OPTIONAL IfcLabel;

```

```

INVERSE

```

```

    ToMaterialProfileSet : IfcMaterialProfileSet FOR MaterialProfiles;

```

```

WHERE

```

```

    NORMALIZEDPRIORITY : NOT(EXISTS(Priority)) OR {0 <= Priority <= 100};

```

```

END_ENTITY;

```

```

ENTITY IfcMaterialProfileWithOffsets

```

```

    SUBTYPE OF(IfcMaterialProfile);

```

```

    OffsetValues : ARRAY [1:2] OF IfcLengthMeasure;

```

```

END_ENTITY;

```

```

ENTITY IfcProfileDef

```

```

    SUPERTYPE OF (ONEOF(IfcArbitraryClosedProfileDef, IfcArbitraryOpenProfileDef,
IfcCompositeProfileDef, IfcDerivedProfileDef, IfcParameterizedProfileDef));

```

```

    ProfileType : IfcProfileTypeEnum;

```

```

    ProfileName : OPTIONAL IfcLabel;

```

```

INVERSE

```

```

    HasExternalReference : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;
    HasProperties         : SET OF IfcProfileProperties FOR ProfileDefinition;
END_ENTITY;

```

```

ENTITY IfcArbitraryClosedProfileDef
  SUBTYPE OF(IfcProfileDef);
  OuterCurve : IfcCurve;
  WHERE
    WR1 : OuterCurve.Dim = 2;
    WR2 : NOT('IFC4.IFCLINE' IN TYPEOF(OuterCurve));
    WR3 : NOT('IFC4.IFCOFFSETCURVE2D' IN TYPEOF(OuterCurve));
END_ENTITY;

```

```

ENTITY IfcArbitraryProfileDefWithVoids
  SUBTYPE OF(IfcArbitraryClosedProfileDef);
  InnerCurves : SET [1:?] OF IfcCurve;
  WHERE
    WR1 : SELF.IfcProfileDef.ProfileType = AREA;
    WR2 : SIZEOF(QUERY(temp <* InnerCurves | temp.Dim <> 2)) = 0;
    WR3 : SIZEOF(QUERY(temp <* InnerCurves | 'IFC4.IFCLINE' IN TYPEOF(temp))) = 0;
END_ENTITY;

```

```

ENTITY IfcCurve
  ABSTRACT SUPERTYPE OF (ONEOF(IfcBoundedCurve, IfcConic, IfcLine, IfcOffsetCurve2D,
IfcOffsetCurve3D, IfcPcurve))
  SUBTYPE OF(IfcGeometricRepresentationItem);
  DERIVE
    Dim : IfcDimensionCount := IfcCurveDim(SELF);
END_ENTITY;

```

```

ENTITY IfcBoundedCurve
  ABSTRACT SUPERTYPE OF (ONEOF(IfcBSplineCurve, IfcCompositeCurve, IfcIndexedPolyCurve,
IfcPolyline, IfcTrimmedCurve))
  SUBTYPE OF(IfcCurve);
END_ENTITY;

```

```

ENTITY IfcBSplineCurve
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcBoundedCurve);
  Degree : IfcInteger;
  ControlPointsList : LIST [2:?] OF IfcCartesianPoint;
  CurveForm : IfcBSplineCurveForm;
  ClosedCurve : IfcLogical;
  SelfIntersect : IfcLogical;
  DERIVE
    UpperIndexOnControlPoints : IfcInteger := (SIZEOF(ControlPointsList) - 1);
    ControlPoints : ARRAY [0:100] OF IfcCartesianPoint :=
IfcListToArray(ControlPointsList,0,UpperIndexOnControlPoints);
  WHERE
    SAMEDIM : SIZEOF(QUERY(Temp <* ControlPointsList |
      Temp.Dim <> ControlPointsList[1].Dim))
      = 0;
END_ENTITY;

```

```

ENTITY IfcBSplineCurveWithKnots
  SUBTYPE OF(IfcBSplineCurve);
  KnotMultiplicities : LIST [2:?] OF IfcInteger;
  Knots : LIST [2:?] OF IfcParameterValue;
  KnotSpec : IfcKnotType;
  DERIVE

```

UpperIndexOnKnots : IfcInteger := SIZEOF(Knots);

WHERE

CONSISTENTBSPLINE : IfcConstraintsParamBSpline(Degree, UpperIndexOnKnots,
UpperIndexOnControlPoints, KnotMultiplicities, Knots);

CORRESPONDINGKNOTLISTS : SIZEOF(KnotMultiplicities) = UpperIndexOnKnots;

END_ENTITY;

ENTITY IfcRationalBSplineCurveWithKnots

SUBTYPE OF(IfcBSplineCurveWithKnots);

WeightsData : LIST [2:?] OF IfcReal;

DERIVE

Weights : ARRAY [0:100] OF IfcReal :=
IfcListToArray(WeightsData,0,SELFIfcBSplineCurve.UpperIndexOnControlPoints);

WHERE

SAMENUMOFWEIGHTSANDPOINTS : SIZEOF(WeightsData) =
SIZEOF(SELFIfcBSplineCurve.ControlPointsList);

WEIGHTSGREATERZERO : IfcCurveWeightsPositive(SELF);

END_ENTITY;

ENTITY IfcCartesianPoint

SUBTYPE OF(IfcPoint);

Coordinates : LIST [1:3] OF IfcLengthMeasure;

DERIVE

Dim : IfcDimensionCount := HIINDEX(Coordinates);

WHERE

CP2DOR3D : HIINDEX(Coordinates) >= 2;

END_ENTITY;

ENTITY IfcPoint

ABSTRACT SUPERTYPE OF (ONEOF(IfcCartesianPoint, IfcPointOnCurve, IfcPointOnSurface))

```
    SUBTYPE OF(IfcGeometricRepresentationItem);  
END_ENTITY;
```

```
ENTITY IfcPointOnCurve  
    SUBTYPE OF(IfcPoint);  
        BasisCurve      : IfcCurve;  
        PointParameter  : IfcParameterValue;  
    DERIVE  
        Dim              : IfcDimensionCount := BasisCurve.Dim;  
END_ENTITY;
```

```
ENTITY IfcPointOnSurface  
    SUBTYPE OF(IfcPoint);  
        BasisSurface    : IfcSurface;  
        PointParameterU : IfcParameterValue;  
        PointParameterV : IfcParameterValue;  
    DERIVE  
        Dim              : IfcDimensionCount := BasisSurface.Dim;  
END_ENTITY;
```

```
ENTITY IfcSurface  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcBoundedSurface, IfcElementarySurface, IfcSweptSurface))  
    SUBTYPE OF(IfcGeometricRepresentationItem);  
    DERIVE  
        Dim : IfcDimensionCount := 3;  
END_ENTITY;
```

```
ENTITY IfcBoundedSurface  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcBSplineSurface, IfcCurveBoundedPlane, IfcCurveBoundedSurface,  
IfcRectangularTrimmedSurface))
```

```
SUBTYPE OF(IfcSurface);  
END_ENTITY;
```

```
ENTITY IfcBSplineSurface
```

```
ABSTRACT SUPERTYPE
```

```
SUBTYPE OF(IfcBoundedSurface);
```

```
UDegree          : IfcInteger;
```

```
VDegree          : IfcInteger;
```

```
ControlPointsList : LIST [2:?] OF LIST [2:?] OF IfcCartesianPoint;
```

```
SurfaceForm      : IfcBSplineSurfaceForm;
```

```
UClosed          : IfcLogical;
```

```
VClosed          : IfcLogical;
```

```
SelfIntersect    : IfcLogical;
```

```
DERIVE
```

```
UUpper           : IfcInteger := SIZEOF(ControlPointsList) - 1;
```

```
VUpper           : IfcInteger := SIZEOF(ControlPointsList[1]) - 1;
```

```
ControlPoints    : ARRAY [0:100] OF ARRAY [0:100] OF IfcCartesianPoint :=  
IfcMakeArrayOfArray(ControlPointsList,  
                     0,UUpper,0,VUpper);
```

```
END_ENTITY;
```

```
ENTITY IfcBSplineSurfaceWithKnots
```

```
SUBTYPE OF(IfcBSplineSurface);
```

```
UMultiplicities : LIST [2:?] OF IfcInteger;
```

```
VMultiplicities : LIST [2:?] OF IfcInteger;
```

```
UKnots          : LIST [2:?] OF IfcParameterValue;
```

```
VKnots          : LIST [2:?] OF IfcParameterValue;
```

```
KnotSpec        : IfcKnotType;
```

```
DERIVE
```

```
KnotVUpper      : IfcInteger := SIZEOF(VKnots);
```

KnotUpper : IfcInteger := SIZEOF(UKnots);

WHERE

UDIRECTIONCONSTRAINTS : IfcConstraintsParamBSpline (
 SELFIfcBSplineSurface.UDegree, KnotUpper,
 SELFIfcBSplineSurface.UUpper, UMultiplcities, UKnots);

VDIRECTIONCONSTRAINTS : IfcConstraintsParamBSpline (
 SELFIfcBSplineSurface.VDegree, KnotVUpper,
 SELFIfcBSplineSurface.VUpper, VMultiplcities, VKnots);

CORRESPONDINGULISTS : SIZEOF(UMultiplcities) = KnotUpper;

CORRESPONDINGVLISTS : SIZEOF(VMultiplcities) = KnotVUpper;

END_ENTITY;

ENTITY IfcRationalBSplineSurfaceWithKnots

SUBTYPE OF(IfcBSplineSurfaceWithKnots);

WeightsData : LIST [2:?] OF LIST [2:?] OF IfcReal;

DERIVE

Weights : ARRAY [0:100] OF ARRAY [0:100] OF IfcReal :=
IfcMakeArrayOfArray(WeightsData,0,UUpper,0,VUpper);

WHERE

CORRESPONDINGWEIGHTSDATALISTS : (SIZEOF(WeightsData) =
SIZEOF(SELFIfcBSplineSurface.ControlPointsList))

AND

(SIZEOF(WeightsData[1]) =
SIZEOF(SELFIfcBSplineSurface.ControlPointsList[1]));

WEIGHTVALUESGREATERZERO : IfcSurfaceWeightsPositive(SELF);

END_ENTITY;

ENTITY IfcCurveBoundedPlane

SUBTYPE OF(IfcBoundedSurface);

BasisSurface : IfcPlane;

OuterBoundary : IfcCurve;

```
    InnerBoundaries : SET OF IfcCurve;  
END_ENTITY;
```

```
ENTITY IfcPlane  
    SUBTYPE OF(IfcElementarySurface);  
END_ENTITY;
```

```
ENTITY IfcElementarySurface  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcCylindricalSurface, IfcPlane))  
    SUBTYPE OF(IfcSurface);  
    Position : IfcAxis2Placement3D;  
END_ENTITY;
```

```
ENTITY IfcCylindricalSurface  
    SUBTYPE OF(IfcElementarySurface);  
    Radius : IfcPositiveLengthMeasure;  
END_ENTITY;
```

```
ENTITY IfcAxis2Placement3D  
    SUBTYPE OF(IfcPlacement);  
    Axis      : OPTIONAL IfcDirection;  
    RefDirection : OPTIONAL IfcDirection;  
    DERIVE  
    P          : LIST [3:3] OF IfcDirection := IfcBuildAxes(Axis, RefDirection);  
    WHERE  
    LOCATIONIS3D      : SELF.IfPlacement.Location.Dim = 3;  
    AXISIS3D          : (NOT (EXISTS (Axis))) OR (Axis.Dim = 3);  
    REFDIRIS3D        : (NOT (EXISTS (RefDirection))) OR (RefDirection.Dim = 3);  
    AXISTOREFDIRPOSITION : (NOT (EXISTS (Axis))) OR (NOT (EXISTS (RefDirection))) OR  
(IfcCrossProduct(Axis,RefDirection).Magnitude > 0.0);
```



```
    AXISANDREFDIRPROVISION : NOT ((EXISTS (Axis)) XOR (EXISTS (RefDirection)));  
END_ENTITY;
```

```
ENTITY IfcDirection
```

```
    SUBTYPE OF(IfcGeometricRepresentationItem);
```

```
    DirectionRatios : LIST [2:3] OF IfcReal;
```

```
    DERIVE
```

```
    Dim          : IfcDimensionCount := HINDEX(DirectionRatios);
```

```
    WHERE
```

```
    MAGNITUDEGREATERZERO : SIZEOF(QUERY(Tmp <* DirectionRatios | Tmp <> 0.0)) > 0;
```

```
END_ENTITY;
```

```
ENTITY IfcCurveBoundedSurface
```

```
    SUBTYPE OF(IfcBoundedSurface);
```

```
    BasisSurface : IfcSurface;
```

```
    Boundaries   : SET [1:?] OF IfcBoundaryCurve;
```

```
    ImplicitOuter : IfcBoolean;
```

```
END_ENTITY;
```

```
ENTITY IfcBoundaryCurve
```

```
    SUBTYPE OF(IfcCompositeCurveOnSurface);
```

```
    WHERE
```

```
    ISCLOSED : SELF=IfcCompositeCurve.ClosedCurve;
```

```
END_ENTITY;
```

```
ENTITY IfcCompositeCurveOnSurface
```

```
    SUBTYPE OF(IfcCompositeCurve);
```

```
    DERIVE
```

```
    BasisSurface : SET [0:1] OF IfcSurface := IfcGetBasisSurface(SELF);
```

```
    WHERE
```

```
    SAMESURFACE : SIZEOF(BasisSurface) > 0;  
END_ENTITY;
```

```
ENTITY IfcCompositeCurve
```

```
    SUBTYPE OF(IfcBoundedCurve);
```

```
    Segments      : LIST [1:?] OF IfcCompositeCurveSegment;
```

```
    SelfIntersect : IfcLogical;
```

```
    DERIVE
```

```
    NSegments     : IfcInteger := SIZEOF(Segments);
```

```
    ClosedCurve   : IfcLogical := Segments[NSegments].Transition <> Discontinuous;
```

```
    WHERE
```

```
        CURVECONTINUOUS : ((NOT ClosedCurve) AND (SIZEOF(QUERY(Temp < * Segments | Temp.Transition =  
Discontinuous)) = 1)) OR ((ClosedCurve) AND (SIZEOF(QUERY(Temp < * Segments | Temp.Transition =  
Discontinuous)) = 0));
```

```
        SAMEDIM         : SIZEOF( QUERY( Temp < * Segments | Temp.Dim <> Segments[1].Dim)) = 0;
```

```
END_ENTITY;
```

```
ENTITY IfcCompositeCurveSegment
```

```
    SUBTYPE OF(IfcGeometricRepresentationItem);
```

```
    Transition     : IfcTransitionCode;
```

```
    SameSense      : IfcBoolean;
```

```
    ParentCurve    : IfcCurve;
```

```
    DERIVE
```

```
    Dim            : IfcDimensionCount := ParentCurve.Dim;
```

```
    INVERSE
```

```
    UsingCurves  : SET [1:?] OF IfcCompositeCurve FOR Segments;
```

```
    WHERE
```

```
        PARENTISBOUNDEDCURVE : ('IFC4.IFCBOUNDEDCURVE' IN TYPEOF(ParentCurve));
```

```
END_ENTITY;
```

```

ENTITY IfcReparametrisedCompositeCurveSegment
  SUBTYPE OF(IfcCompositeCurveSegment);
  ParamLength : IfcParameterValue;
  WHERE
    POSITIVELENGTHPARAMETER : ParamLength > 0.0;
END_ENTITY;

```

```

ENTITY IfcOuterBoundaryCurve
  SUBTYPE OF(IfcBoundaryCurve);
END_ENTITY;

```

```

ENTITY IfcRectangularTrimmedSurface
  SUBTYPE OF(IfcBoundedSurface);
  BasisSurface : IfcSurface;
  U1           : IfcParameterValue;
  V1           : IfcParameterValue;
  U2           : IfcParameterValue;
  V2           : IfcParameterValue;
  Usense       : IfcBoolean;
  Vsense       : IfcBoolean;
  WHERE
    U1ANDU2DIFFERENT : U1 <> U2;
    V1ANDV2DIFFERENT : V1 <> V2;
    USENSECOMPATIBLE : (('IFC4.IFCELEMENTARYSURFACE' IN TYPEOF(BasisSurface)) AND
      (NOT ('IFC4.IFCPLANE' IN TYPEOF(BasisSurface)))) OR
      ('IFC4.IFCSURFACEOFREVOLUTION' IN TYPEOF(BasisSurface)) OR
      (Usense = (U2 > U1));
    VSENSECOMPATIBLE : Vsense = (V2 > V1);
END_ENTITY;

```

```

ENTITY IfcSweptSurface
  ABSTRACT SUPERTYPE OF (ONEOF(IfcSurfaceOfLinearExtrusion, IfcSurfaceOfRevolution))
  SUBTYPE OF(IfcSurface);
  SweptCurve : IfcProfileDef;
  Position    : OPTIONAL IfcAxis2Placement3D;
  WHERE
    SWEPTCURVETYPE : SweptCurve.ProfileType = IfcProfileTypeEnum.Curve;
END_ENTITY;

```

```

ENTITY IfcSurfaceOfLinearExtrusion
  SUBTYPE OF(IfcSweptSurface);
  ExtrudedDirection : IfcDirection;
  Depth              : IfcLengthMeasure;
  DERIVE
    ExtrusionAxis    : IfcVector := IfcRepresentationItem() || IfcGeometricRepresentationItem
    () || IfcVector (ExtrudedDirection, Depth);
  WHERE
    DEPTHGREATERZERO : Depth > 0.;
END_ENTITY;

```

```

ENTITY IfcVector
  SUBTYPE OF(IfcGeometricRepresentationItem);
  Orientation : IfcDirection;
  Magnitude   : IfcLengthMeasure;
  DERIVE
    Dim          : IfcDimensionCount := Orientation.Dim;
  WHERE
    MAGGREATEROREQUALZERO : Magnitude >= 0.0;
END_ENTITY;

```

```

ENTITY IfcSurfaceOfRevolution
  SUBTYPE OF(IfcSweptSurface);
  AxisPosition : IfcAxis1Placement;
  DERIVE
    AxisLine      : IfcLine := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcCurve() || IfcLine(AxisPosition.Location,
                        IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcVector(AxisPosition.Z,1.0));
  END_ENTITY;

```

```

ENTITY IfcAxis1Placement
  SUBTYPE OF(IfcPlacement);
  Axis : OPTIONAL IfcDirection;
  DERIVE
    Z      : IfcDirection := NVL (IfcNormalise(Axis), IfcRepresentationItem() ||
IfcGeometricRepresentationItem () || IfcDirection([0.0,0.0,1.0]));
  WHERE
    AXISIS3D      : (NOT (EXISTS (Axis))) OR (Axis.Dim = 3);
    LOCATIONIS3D : SELFIfcPlacement.Location.Dim = 3;
  END_ENTITY;

```

```

ENTITY IfcLine
  SUBTYPE OF(IfcCurve);
  Pnt : IfcCartesianPoint;
  Dir : IfcVector;
  WHERE
    SAMEDIM : Dir.Dim = Pnt.Dim;
  END_ENTITY;

```

```

ENTITY IfcIndexedPolyCurve
  SUBTYPE OF(IfcBoundedCurve);

```

```

Points      : IfcCartesianPointList;
Segments    : OPTIONAL LIST [1:?] OF IfcSegmentIndexSelect;
SelfIntersect : OPTIONAL IfcBoolean;

WHERE

    CONSECUTIVE : (SIZEOF(Segments) = 0) OR IfcConsecutiveSegments(Segments);
END_ENTITY;

```

```

ENTITY IfcCartesianPointList
    ABSTRACT SUPERTYPE OF (ONEOF(IfcCartesianPointList2D, IfcCartesianPointList3D))
    SUBTYPE OF(IfcGeometricRepresentationItem);
    DERIVE
        Dim : IfcDimensionCount := IfcPointListDim(SELF);
END_ENTITY;

```

```

ENTITY IfcCartesianPointList2D
    SUBTYPE OF(IfcCartesianPointList);
    CoordList : LIST [1:?] OF LIST [2:2] OF IfcLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcCartesianPointList3D
    SUBTYPE OF(IfcCartesianPointList);
    CoordList : LIST [1:?] OF LIST [3:3] OF IfcLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcPolyline
    SUBTYPE OF(IfcBoundedCurve);
    Points : LIST [2:?] OF IfcCartesianPoint;
    WHERE
        SAMEDIM : SIZEOF(QUERY(Temp <* Points | Temp.Dim <> Points[1].Dim)) = 0;
END_ENTITY;

```

```

ENTITY IfcTrimmedCurve
  SUBTYPE OF(IfcBoundedCurve);
  BasisCurve      : IfcCurve;
  Trim1           : SET [1:2] OF IfcTrimmingSelect;
  Trim2           : SET [1:2] OF IfcTrimmingSelect;
  SenseAgreement  : IfcBoolean;
  MasterRepresentation : IfcTrimmingPreference;

  WHERE
    TRIM1VALUESCONSISTENT : (HIINDEX(Trim1) = 1) OR (TYPEOF(Trim1[1]) <> TYPEOF(Trim1[2]));
    TRIM2VALUESCONSISTENT : (HIINDEX(Trim2) = 1) OR (TYPEOF(Trim2[1]) <> TYPEOF(Trim2[2]));
    NOTRIMMOFBOUNDEDCURVES : NOT('IFC4.IFCBOUNDEDCURVE' IN TYPEOF(BasisCurve));
END_ENTITY;

```

```

ENTITY IfcConic
  ABSTRACT SUPERTYPE OF (ONEOF(IfcCircle, IfcEllipse))
  SUBTYPE OF(IfcCurve);
  Position : IfcAxis2Placement;
END_ENTITY;

```

```

ENTITY IfcCircle
  SUBTYPE OF(IfcConic);
  Radius : IfcPositiveLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcEllipse
  SUBTYPE OF(IfcConic);
  SemiAxis1 : IfcPositiveLengthMeasure;
  SemiAxis2 : IfcPositiveLengthMeasure;
END_ENTITY;

```

```
ENTITY IfcOffsetCurve2D
  SUBTYPE OF(IfcCurve);
  BasisCurve      : IfcCurve;
  Distance        : IfcLengthMeasure;
  SelfIntersect  : IfcLogical;
  WHERE
    DIMS2D : BasisCurve.Dim = 2;
END_ENTITY;
```

```
ENTITY IfcOffsetCurve3D
  SUBTYPE OF(IfcCurve);
  BasisCurve      : IfcCurve;
  Distance        : IfcLengthMeasure;
  SelfIntersect  : IfcLogical;
  RefDirection   : IfcDirection;
  WHERE
    DIMS3D : BasisCurve.Dim = 3;
END_ENTITY;
```

```
ENTITY IfcPcurve
  SUBTYPE OF(IfcCurve);
  BasisSurface    : IfcSurface;
  ReferenceCurve  : IfcCurve;
  WHERE
    DIMS2D : ReferenceCurve.Dim = 2;
END_ENTITY;
```

```
ENTITY IfcArbitraryOpenProfileDef
  SUBTYPE OF(IfcProfileDef);
```


Curve : IfcBoundedCurve;

WHERE

WR11 : ('IFC4.IFCCENTERLINEPROFILEDEF' IN TYPEOF(SELF)) OR

(SELFIfcProfileDef.ProfileType = IfcProfileTypeEnum.CURVE);

WR12 : Curve.Dim = 2;

END_ENTITY;

ENTITY IfcCenterLineProfileDef

SUBTYPE OF(IfcArbitraryOpenProfileDef);

Thickness : IfcPositiveLengthMeasure;

END_ENTITY;

ENTITY IfcCompositeProfileDef

SUBTYPE OF(IfcProfileDef);

Profiles : SET [2:?] OF IfcProfileDef;

Label : OPTIONAL IfcLabel;

WHERE

INVARIANTPROFILETYPE : SIZEOF(QUERY(temp <* Profiles | temp.ProfileType <>
Profiles[1].ProfileType)) = 0;

NORECURSION : SIZEOF(QUERY(temp <* Profiles | 'IFC4.IFCCOMPOSITEPROFILEDEF' IN
TYPEOF(temp))) = 0;

END_ENTITY;

ENTITY IfcDerivedProfileDef

SUBTYPE OF(IfcProfileDef);

ParentProfile : IfcProfileDef;

Operator : IfcCartesianTransformationOperator2D;

Label : OPTIONAL IfcLabel;

WHERE

INVARIANTPROFILETYPE : SELFWIfcProfileDef.ProfileType = ParentProfile.ProfileType;

END_ENTITY;

ENTITY IfcMirroredProfileDef

SUBTYPE OF(IfcDerivedProfileDef);

DERIVE

SELFWIfcDerivedProfileDef.Operator : IfcCartesianTransformationOperator2D :=

IfcRepresentationItem() ||

IfcGeometricRepresentationItem() ||

IfcCartesianTransformationOperator(

-- Axis1

IfcRepresentationItem() ||

IfcGeometricRepresentationItem() ||

IfcDirection([-1., 0.]),

-- Axis2

IfcRepresentationItem() ||

IfcGeometricRepresentationItem() ||

IfcDirection([0., 1.]),

-- LocalOrigin

IfcRepresentationItem() ||

IfcGeometricRepresentationItem() ||

IfcPoint() || IfcCartesianPoint([0., 0.]),

-- Scale

1.) ||

IfcCartesianTransformationOperator2D();

END_ENTITY;

ENTITY IfcCartesianTransformationOperator2D

SUBTYPE OF(IfcCartesianTransformationOperator);

DERIVE

U : LIST [2:2] OF IfcDirection :=

IfcBaseAxis(2,SELFWIfcCartesianTransformationOperator.Axis1,

```
SELFWIfcCartesianTransformationOperator.Axis2,?);
```

```
WHERE
```

```
DIMEQUAL2 : SELFWIfcCartesianTransformationOperator.Dim = 2;
```

```
AXIS1IS2D : NOT(EXISTS(SELFWIfcCartesianTransformationOperator.Axis1)) OR  
(SELFWIfcCartesianTransformationOperator.Axis1.Dim = 2);
```

```
AXIS2IS2D : NOT(EXISTS(SELFWIfcCartesianTransformationOperator.Axis2)) OR  
(SELFWIfcCartesianTransformationOperator.Axis2.Dim = 2);
```

```
END_ENTITY;
```

```
ENTITY IfcCartesianTransformationOperator
```

```
ABSTRACT SUPERTYPE OF (ONEOF(IfcCartesianTransformationOperator2D,  
IfcCartesianTransformationOperator3D))
```

```
SUBTYPE OF(IfcGeometricRepresentationItem);
```

```
Axis1 : OPTIONAL IfcDirection;
```

```
Axis2 : OPTIONAL IfcDirection;
```

```
LocalOrigin : IfcCartesianPoint;
```

```
Scale : OPTIONAL IfcReal;
```

```
DERIVE
```

```
Scale : IfcReal := NVL(Scale, 1.0);
```

```
Dim : IfcDimensionCount := LocalOrigin.Dim;
```

```
WHERE
```

```
SCALEGREATERZERO : Scale > 0.0;
```

```
END_ENTITY;
```

```
ENTITY IfcCartesianTransformationOperator3D
```

```
SUBTYPE OF(IfcCartesianTransformationOperator);
```

```
Axis3 : OPTIONAL IfcDirection;
```

```
DERIVE
```

```
U : LIST [3:3] OF IfcDirection :=  
IfcBaseAxis(3,SELFWIfcCartesianTransformationOperator.Axis1,
```

```

        SELFWIfcCartesianTransformationOperator.Axis2,Axis3);
WHERE
    DIMIS3D : SELFWIfcCartesianTransformationOperator.Dim = 3;
    AXIS1IS3D : NOT(EXISTS(SELFWIfcCartesianTransformationOperator.Axis1)) OR
        (SELFWIfcCartesianTransformationOperator.Axis1.Dim = 3);
    AXIS2IS3D : NOT(EXISTS(SELFWIfcCartesianTransformationOperator.Axis2)) OR
        (SELFWIfcCartesianTransformationOperator.Axis2.Dim = 3);
    AXIS3IS3D : NOT(EXISTS(Axis3)) OR (Axis3.Dim = 3);
END_ENTITY;

ENTITY IfcCartesianTransformationOperator3DnonUniform
    SUBTYPE OF(IfcCartesianTransformationOperator3D);
    Scale2 : OPTIONAL IfcReal;
    Scale3 : OPTIONAL IfcReal;
    DERIVE
        ScI2 : IfcReal := NVL(Scale2, SELFWIfcCartesianTransformationOperator.Scl);
        ScI3 : IfcReal := NVL(Scale3, SELFWIfcCartesianTransformationOperator.Scl);
    WHERE
        SCALE2GREATERZERO : ScI2 > 0.0;
        SCALE3GREATERZERO : ScI3 > 0.0;
END_ENTITY;

ENTITY IfcCartesianTransformationOperator2DnonUniform
    SUBTYPE OF(IfcCartesianTransformationOperator2D);
    Scale2 : OPTIONAL IfcReal;
    DERIVE
        ScI2 : IfcReal := NVL(Scale2, SELFWIfcCartesianTransformationOperator.Scl);
    WHERE
        SCALE2GREATERZERO : ScI2 > 0.0;
END_ENTITY;

```

ENTITY IfcParameterizedProfileDef

ABSTRACT SUPERTYPE OF (ONEOF(IfcAsymmetricIShapeProfileDef, IfcCShapeProfileDef, IfcCircleProfileDef, IfcEllipseProfileDef, IfcIShapeProfileDef, IfcLShapeProfileDef, IfcRectangleProfileDef, IfcTShapeProfileDef, IfcTrapeziumProfileDef, IfcUShapeProfileDef, IfcZShapeProfileDef))

SUBTYPE OF(IfcProfileDef);

Position : OPTIONAL IfcAxis2Placement2D;

END_ENTITY;

ENTITY IfcAsymmetricIShapeProfileDef

SUBTYPE OF(IfcParameterizedProfileDef);

BottomFlangeWidth : IfcPositiveLengthMeasure;

OverallDepth : IfcPositiveLengthMeasure;

WebThickness : IfcPositiveLengthMeasure;

BottomFlangeThickness : IfcPositiveLengthMeasure;

BottomFlangeFilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;

TopFlangeWidth : IfcPositiveLengthMeasure;

TopFlangeThickness : OPTIONAL IfcPositiveLengthMeasure;

TopFlangeFilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;

BottomFlangeEdgeRadius : OPTIONAL IfcNonNegativeLengthMeasure;

BottomFlangeSlope : OPTIONAL IfcPlaneAngleMeasure;

TopFlangeEdgeRadius : OPTIONAL IfcNonNegativeLengthMeasure;

TopFlangeSlope : OPTIONAL IfcPlaneAngleMeasure;

WHERE

VALIDFLANGETHICKNESS : NOT(EXISTS(TopFlangeThickness)) OR ((BottomFlangeThickness + TopFlangeThickness) < OverallDepth);

VALIDWEBTHICKNESS : (WebThickness < BottomFlangeWidth) AND (WebThickness < TopFlangeWidth);

VALIDBOTTOMFILLETADIUS : (NOT(EXISTS(BottomFlangeFilletRadius))) OR (BottomFlangeFilletRadius <= (BottomFlangeWidth - WebThickness)/2.);

VALIDTOPFILLETRADIUS : (NOT(EXISTS(TopFlangeFilletRadius))) OR
(TopFlangeFilletRadius <= (TopFlangeWidth - WebThickness)/2.);

END_ENTITY;

ENTITY IfcCShapeProfileDef

SUBTYPE OF(IfcParameterizedProfileDef);

Depth : IfcPositiveLengthMeasure;

Width : IfcPositiveLengthMeasure;

WallThickness : IfcPositiveLengthMeasure;

Girth : IfcPositiveLengthMeasure;

InternalFilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;

WHERE

VALIDGIRTH : Girth < (Depth / 2.);

VALIDINTERNALFILLETRADIUS : NOT(EXISTS(InternalFilletRadius)) OR

((InternalFilletRadius <= Width/2. - WallThickness) AND
(InternalFilletRadius <= Depth/2. - WallThickness));

VALIDWALLTHICKNESS : (WallThickness < Width/2.) AND (WallThickness < Depth/2.);

END_ENTITY;

ENTITY IfcCircleProfileDef

SUBTYPE OF(IfcParameterizedProfileDef);

Radius : IfcPositiveLengthMeasure;

END_ENTITY;

ENTITY IfcCircleHollowProfileDef

SUBTYPE OF(IfcCircleProfileDef);

WallThickness : IfcPositiveLengthMeasure;

WHERE

WR1 : WallThickness < SELF.IfCircleProfileDef.Radius;

END_ENTITY;

```

ENTITY IfcEllipseProfileDef
  SUBTYPE OF(IfcParameterizedProfileDef);
  SemiAxis1 : IfcPositiveLengthMeasure;
  SemiAxis2 : IfcPositiveLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcIShapeProfileDef
  SUBTYPE OF(IfcParameterizedProfileDef);
  OverallWidth      : IfcPositiveLengthMeasure;
  OverallDepth      : IfcPositiveLengthMeasure;
  WebThickness       : IfcPositiveLengthMeasure;
  FlangeThickness    : IfcPositiveLengthMeasure;
  FilletRadius       : OPTIONAL IfcNonNegativeLengthMeasure;
  FlangeEdgeRadius   : OPTIONAL IfcNonNegativeLengthMeasure;
  FlangeSlope        : OPTIONAL IfcPlaneAngleMeasure;
  WHERE
  VALIDFLANGETHICKNESS : (2. * FlangeThickness) < OverallDepth;
  VALIDWEBTHICKNESS    : WebThickness < OverallWidth;
  VALIDFILLETADIUS     : NOT(EXISTS(FilletRadius)) OR
                        ((FilletRadius <= (OverallWidth - WebThickness)/2.) AND
                         (FilletRadius <= (OverallDepth - (2. * FlangeThickness))/2.));
END_ENTITY;

```

```

ENTITY IfcLShapeProfileDef
  SUBTYPE OF(IfcParameterizedProfileDef);
  Depth      : IfcPositiveLengthMeasure;
  Width      : OPTIONAL IfcPositiveLengthMeasure;
  Thickness   : IfcPositiveLengthMeasure;
  FilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;

```

EdgeRadius : OPTIONAL IfcNonNegativeLengthMeasure;

LegSlope : OPTIONAL IfcPlaneAngleMeasure;

WHERE

VALIDTHICKNESS : (Thickness < Depth) AND (NOT(EXISTS(Width)) OR (Thickness < Width));

END_ENTITY;

ENTITY IfcRectangleProfileDef

SUPERTYPE OF (ONEOF(IfcRectangleHollowProfileDef, IfcRoundedRectangleProfileDef))

SUBTYPE OF(IfcParameterizedProfileDef);

XDim : IfcPositiveLengthMeasure;

YDim : IfcPositiveLengthMeasure;

END_ENTITY;

ENTITY IfcRectangleHollowProfileDef

SUBTYPE OF(IfcRectangleProfileDef);

WallThickness : IfcPositiveLengthMeasure;

InnerFilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;

OuterFilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;

WHERE

VALIDWALLTHICKNESS : (WallThickness < (SELFIfcRectangleProfileDef.XDim/2.)) AND
(WallThickness < (SELFIfcRectangleProfileDef.YDim/2.));

VALIDINNERRADIUS : NOT(EXISTS(InnerFilletRadius)) OR
((InnerFilletRadius <= (SELFIfcRectangleProfileDef.XDim/2. -
WallThickness)) AND
(InnerFilletRadius <= (SELFIfcRectangleProfileDef.YDim/2. -
WallThickness)));

VALIDOUTERRADIUS : NOT(EXISTS(OuterFilletRadius)) OR
((OuterFilletRadius <= (SELFIfcRectangleProfileDef.XDim/2.)) AND
(OuterFilletRadius <= (SELFIfcRectangleProfileDef.YDim/2.)));

END_ENTITY;


```

ENTITY IfcRoundedRectangleProfileDef
  SUBTYPE OF(IfcRectangleProfileDef);
  RoundingRadius : IfcPositiveLengthMeasure;
  WHERE
    VALIDRADIUS : ((RoundingRadius <= (SELFIfcRectangleProfileDef.XDim/2.)) AND
      (RoundingRadius <= (SELFIfcRectangleProfileDef.YDim/2.)));
END_ENTITY;

```

```

ENTITY IfcTShapeProfileDef
  SUBTYPE OF(IfcParameterizedProfileDef);
  Depth : IfcPositiveLengthMeasure;
  FlangeWidth : IfcPositiveLengthMeasure;
  WebThickness : IfcPositiveLengthMeasure;
  FlangeThickness : IfcPositiveLengthMeasure;
  FilletRadius : OPTIONAL IfcNonNegativeLengthMeasure;
  FlangeEdgeRadius : OPTIONAL IfcNonNegativeLengthMeasure;
  WebEdgeRadius : OPTIONAL IfcNonNegativeLengthMeasure;
  WebSlope : OPTIONAL IfcPlaneAngleMeasure;
  FlangeSlope : OPTIONAL IfcPlaneAngleMeasure;
  WHERE
    VALIDFLANGETHICKNESS : FlangeThickness < Depth;
    VALIDWEBTHICKNESS : WebThickness < FlangeWidth;
END_ENTITY;

```

```

ENTITY IfcTrapeziumProfileDef
  SUBTYPE OF(IfcParameterizedProfileDef);
  BottomXDim : IfcPositiveLengthMeasure;
  TopXDim : IfcPositiveLengthMeasure;
  YDim : IfcPositiveLengthMeasure;

```

```
TopXOffset : IfcLengthMeasure;  
END_ENTITY;
```

```
ENTITY IfcUShapeProfileDef
```

```
  SUBTYPE OF(IfcParameterizedProfileDef);
```

```
    Depth          : IfcPositiveLengthMeasure;  
    FlangeWidth    : IfcPositiveLengthMeasure;  
    WebThickness   : IfcPositiveLengthMeasure;  
    FlangeThickness : IfcPositiveLengthMeasure;  
    FilletRadius   : OPTIONAL IfcNonNegativeLengthMeasure;  
    EdgeRadius     : OPTIONAL IfcNonNegativeLengthMeasure;  
    FlangeSlope    : OPTIONAL IfcPlaneAngleMeasure;
```

```
  WHERE
```

```
    VALIDFLANGETHICKNESS : FlangeThickness < (Depth / 2.);  
    VALIDWEBTHICKNESS    : WebThickness < FlangeWidth;
```

```
END_ENTITY;
```

```
ENTITY IfcZShapeProfileDef
```

```
  SUBTYPE OF(IfcParameterizedProfileDef);
```

```
    Depth          : IfcPositiveLengthMeasure;  
    FlangeWidth    : IfcPositiveLengthMeasure;  
    WebThickness   : IfcPositiveLengthMeasure;  
    FlangeThickness : IfcPositiveLengthMeasure;  
    FilletRadius   : OPTIONAL IfcNonNegativeLengthMeasure;  
    EdgeRadius     : OPTIONAL IfcNonNegativeLengthMeasure;
```

```
  WHERE
```

```
    VALIDFLANGETHICKNESS : FlangeThickness < (Depth / 2.);
```

```
END_ENTITY;
```

```
ENTITY IfcExternalReferenceRelationship
```

```
SUBTYPE OF(IfcResourceLevelRelationship);  
    RelatingReference      : IfcExternalReference;  
    RelatedResourceObjects : SET [1:?] OF IfcResourceObjectSelect;  
END_ENTITY;
```

```
ENTITY IfcResourceLevelRelationship
```

```
    ABSTRACT SUPERTYPE OF (ONEOF(IfcApprovalRelationship, IfcCurrencyRelationship,  
IfcDocumentInformationRelationship, IfcExternalReferenceRelationship, IfcMaterialRelationship,  
IfcOrganizationRelationship, IfcPropertyDependencyRelationship, IfcResourceApprovalRelationship,  
IfcResourceConstraintRelationship));
```

```
    Name      : OPTIONAL IfcLabel;
```

```
    Description : OPTIONAL IfcText;
```

```
END_ENTITY;
```

```
ENTITY IfcApprovalRelationship
```

```
    SUBTYPE OF(IfcResourceLevelRelationship);
```

```
    RelatingApproval : IfcApproval;
```

```
    RelatedApprovals : SET [1:?] OF IfcApproval;
```

```
END_ENTITY;
```

```
ENTITY IfcApproval;
```

```
    Identifier      : OPTIONAL IfcIdentifier;
```

```
    Name            : OPTIONAL IfcLabel;
```

```
    Description     : OPTIONAL IfcText;
```

```
    TimeOfApproval : OPTIONAL IfcDateTime;
```

```
    Status          : OPTIONAL IfcLabel;
```

```
    Level           : OPTIONAL IfcLabel;
```

```
    Qualifier       : OPTIONAL IfcText;
```

```
    RequestingApproval : OPTIONAL IfcActorSelect;
```

```
    GivingApproval  : OPTIONAL IfcActorSelect;
```

```
INVERSE
```

HasExternalReferences : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;
 ApprovedObjects : SET OF IfcRelAssociatesApproval FOR RelatingApproval;
 ApprovedResources : SET OF IfcResourceApprovalRelationship FOR RelatingApproval;
 IsRelatedWith : SET OF IfcApprovalRelationship FOR RelatedApprovals;
 Relates : SET OF IfcApprovalRelationship FOR RelatingApproval;

WHERE

HASIDENTIFIERORNAME : EXISTS (Identifier) OR EXISTS (Name);

END_ENTITY;

ENTITY IfcOrganization;

Identification : OPTIONAL IfcIdentifier;
 Name : IfcLabel;
 Description : OPTIONAL IfcText;
 Roles : OPTIONAL LIST [1:?] OF IfcActorRole;
 Addresses : OPTIONAL LIST [1:?] OF IfcAddress;

INVERSE

IsRelatedBy : SET OF IfcOrganizationRelationship FOR RelatedOrganizations;
 Relates : SET OF IfcOrganizationRelationship FOR RelatingOrganization;
 Engages : SET OF IfcPersonAndOrganization FOR TheOrganization;

END_ENTITY;

ENTITY IfcActorRole;

Role : IfcRoleEnum;
 UserDefinedRole : OPTIONAL IfcLabel;
 Description : OPTIONAL IfcText;

INVERSE

HasExternalReference : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;

WHERE

WR1 : (Role <> IfcRoleEnum.USERDEFINED) OR
 ((Role = IfcRoleEnum.USERDEFINED) AND

```

        EXISTS(SELF.UserDefinedRole));
END_ENTITY;

ENTITY IfcAddress
  ABSTRACT SUPERTYPE OF (ONEOF(IfcPostalAddress, IfcTelecomAddress));
  Purpose          : OPTIONAL IfcAddressTypeEnum;
  Description      : OPTIONAL IfcText;
  UserDefinedPurpose : OPTIONAL IfcLabel;
  INVERSE
    OfPerson       : SET OF IfcPerson FOR Addresses;
    OfOrganization : SET OF IfcOrganization FOR Addresses;
  WHERE
    WR1 : (NOT(EXISTS(Purpose))) OR
          ((Purpose <> IfcAddressTypeEnum.USERDEFINED) OR
           ((Purpose = IfcAddressTypeEnum.USERDEFINED) AND
            EXISTS(SELF.UserDefinedPurpose)));
END_ENTITY;

```

```

ENTITY IfcPostalAddress
  SUBTYPE OF(IfcAddress);
  InternalLocation : OPTIONAL IfcLabel;
  AddressLines    : OPTIONAL LIST [1:?] OF IfcLabel;
  PostalBox       : OPTIONAL IfcLabel;
  Town            : OPTIONAL IfcLabel;
  Region         : OPTIONAL IfcLabel;
  PostalCode     : OPTIONAL IfcLabel;
  Country        : OPTIONAL IfcLabel;
  WHERE
    WR1 : EXISTS (InternalLocation) OR
          EXISTS (AddressLines) OR

```

EXISTS (PostalBox) OR
EXISTS (PostalCode) OR
EXISTS (Town) OR
EXISTS (Region) OR
EXISTS (Country);

END_ENTITY;

ENTITY IfcTelecomAddress

SUBTYPE OF(IfcAddress);

TelephoneNumbers : OPTIONAL LIST [1:?] OF IfcLabel;
FacsimileNumbers : OPTIONAL LIST [1:?] OF IfcLabel;
PagerNumber : OPTIONAL IfcLabel;
ElectronicMailAddresses : OPTIONAL LIST [1:?] OF IfcLabel;
WWWHomePageURL : OPTIONAL IfcURIReference;
MessagingIDs : OPTIONAL LIST [1:?] OF IfcURIReference;

WHERE

MINIMUMDATAPROVIDED : EXISTS (TelephoneNumbers) OR
EXISTS (FacsimileNumbers) OR
EXISTS (PagerNumber) OR
EXISTS (ElectronicMailAddresses) OR
EXISTS (WWWHomePageURL) OR
EXISTS (MessagingIDs);

END_ENTITY;

ENTITY IfcPerson;

Identification : OPTIONAL IfcIdentifier;
FamilyName : OPTIONAL IfcLabel;
GivenName : OPTIONAL IfcLabel;
MiddleNames : OPTIONAL LIST [1:?] OF IfcLabel;
PrefixTitles : OPTIONAL LIST [1:?] OF IfcLabel;

SuffixTitles : OPTIONAL LIST [1:?] OF IfcLabel;
Roles : OPTIONAL LIST [1:?] OF IfcActorRole;
Addresses : OPTIONAL LIST [1:?] OF IfcAddress;

INVERSE

EngagedIn : SET OF IfcPersonAndOrganization FOR ThePerson;

WHERE

IDENTIFIABLEPERSON : EXISTS(Identification) OR EXISTS(FamilyName) OR EXISTS(GivenName);

VALIDSETOFNAMES : NOT EXISTS(MiddleNames) OR EXISTS(FamilyName) OR EXISTS(GivenName);

END_ENTITY;

ENTITY IfcPersonAndOrganization;

ThePerson : IfcPerson;

TheOrganization : IfcOrganization;

Roles : OPTIONAL LIST [1:?] OF IfcActorRole;

END_ENTITY;

ENTITY IfcOrganizationRelationship

SUBTYPE OF(IfcResourceLevelRelationship);

RelatingOrganization : IfcOrganization;

RelatedOrganizations : SET [1:?] OF IfcOrganization;

END_ENTITY;

ENTITY IfcRelAssociatesApproval

SUBTYPE OF(IfcRelAssociates);

RelatingApproval : IfcApproval;

END_ENTITY;

ENTITY IfcRelAssociates

ABSTRACT SUPERTYPE OF (ONEOF(IfcRelAssociatesApproval, IfcRelAssociatesClassification,
IfcRelAssociatesConstraint, IfcRelAssociatesDocument, IfcRelAssociatesLibrary,

IfcRelAssociatesMaterial))

SUBTYPE OF(IfcRelationship);

RelatedObjects : SET [1:?] OF IfcDefinitionSelect;

END_ENTITY;

ENTITY IfcRelationship

ABSTRACT SUPERTYPE OF (ONEOF(IfcRelAssigns, IfcRelAssociates, IfcRelConnects, IfcRelDeclares, IfcRelDecomposes, IfcRelDefines))

SUBTYPE OF(IfcRoot);

END_ENTITY;

ENTITY IfcRelAssigns

ABSTRACT SUPERTYPE OF (ONEOF(IfcRelAssignsToActor, IfcRelAssignsToControl, IfcRelAssignsToGroup, IfcRelAssignsToProcess, IfcRelAssignsToProduct, IfcRelAssignsToResource))

SUBTYPE OF(IfcRelationship);

RelatedObjects : SET [1:?] OF IfcObjectDefinition;

RelatedObjectsType : OPTIONAL IfcObjectTypeEnum;

WHERE

WR1 : IfcCorrectObjectAssignment(RelatedObjectsType, RelatedObjects);

END_ENTITY;

ENTITY IfcRelAssignsToActor

SUBTYPE OF(IfcRelAssigns);

RelatingActor : IfcActor;

ActingRole : OPTIONAL IfcActorRole;

WHERE

NOSELFREFERENCE : SIZEOF(QUERY(Temp <* SELF IfcRelAssigns.RelatedObjects | RelatingActor :=: Temp)) = 0;

END_ENTITY;

ENTITY IfcActor

SUBTYPE OF(IfcObject);

TheActor : IfcActorSelect;

INVERSE

IsActingUpon : SET OF IfcRelAssignsToActor FOR RelatingActor;

END_ENTITY;

ENTITY IfcObject

ABSTRACT SUPERTYPE OF (ONEOF(IfcActor, IfcControl, IfcGroup, IfcProcess, IfcProduct, IfcResource))

SUBTYPE OF(IfcObjectDefinition);

ObjectType : OPTIONAL IfcLabel;

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;

WHERE

UNIQUEPROPERTYSETNAMES : ((SIZEOF(IsDefinedBy) = 0) OR IfcUniqueDefinitionNames(IsDefinedBy));

END_ENTITY;

ENTITY IfcControl

ABSTRACT SUPERTYPE OF (ONEOF(IfcActionRequest, IfcCostItem, IfcCostSchedule, IfcPerformanceHistory, IfcPermit, IfcProjectOrder, IfcWorkCalendar, IfcWorkControl))

SUBTYPE OF(IfcObject);

Identification : OPTIONAL IfcIdentifier;

INVERSE

Controls : SET OF IfcRelAssignsToControl FOR RelatingControl;

END_ENTITY;

ENTITY IfcActionRequest

```
SUBTYPE OF(IfcControl);
    PredefinedType : OPTIONAL IfcActionRequestTypeEnum;
    Status          : OPTIONAL IfcLabel;
    LongDescription : OPTIONAL IfcText;
END_ENTITY;
```

```
ENTITY IfcCostItem
    SUBTYPE OF(IfcControl);
    PredefinedType : OPTIONAL IfcCostItemTypeEnum;
    CostValues     : OPTIONAL LIST [1:?] OF IfcCostValue;
    CostQuantities : OPTIONAL LIST [1:?] OF IfcPhysicalQuantity;
END_ENTITY;
```

```
ENTITY IfcCostValue
    SUBTYPE OF(IfcAppliedValue);
END_ENTITY;
```

```
ENTITY IfcAppliedValue;
    Name          : OPTIONAL IfcLabel;
    Description    : OPTIONAL IfcText;
    AppliedValue  : OPTIONAL IfcAppliedValueSelect;
    UnitBasis     : OPTIONAL IfcMeasureWithUnit;
    ApplicableDate : OPTIONAL IfcDate;
    FixedUntilDate : OPTIONAL IfcDate;
    Category      : OPTIONAL IfcLabel;
    Condition     : OPTIONAL IfcLabel;
    ArithmeticOperator : OPTIONAL IfcArithmeticOperatorEnum;
    Components    : OPTIONAL LIST [1:?] OF IfcAppliedValue;
INVERSE
    HasExternalReference : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;
```

END_ENTITY;

ENTITY IfcMeasureWithUnit;

ValueComponent : IfcValue;

UnitComponent : IfcUnit;

END_ENTITY;

ENTITY IfcDerivedUnit;

Elements : SET [1:?] OF IfcDerivedUnitElement;

UnitType : IfcDerivedUnitEnum;

UserDefinedType : OPTIONAL IfcLabel;

DERIVE

Dimensions : IfcDimensionalExponents := IfcDerivedDimensionalExponents(Elements);

WHERE

WR1 : (SIZEOF (Elements) > 1) OR ((SIZEOF (Elements) = 1) AND (Elements[1].Exponent < 1));

WR2 : (UnitType < IfcDerivedUnitEnum.USERDEFINED) OR

((UnitType = IfcDerivedUnitEnum.USERDEFINED) AND

(EXISTS(SELF.UserDefinedType)));

END_ENTITY;

ENTITY IfcDerivedUnitElement;

Unit : IfcNamedUnit;

Exponent : INTEGER;

END_ENTITY;

ENTITY IfcNamedUnit

ABSTRACT SUPERTYPE OF (ONEOF(IfcContextDependentUnit, IfcConversionBasedUnit, IfcSIUnit));

Dimensions : IfcDimensionalExponents;

UnitType : IfcUnitEnum;

WHERE

```
WR1 : IfcCorrectDimensions (SELF.UnitType, SELF.Dimensions);  
END_ENTITY;
```

```
ENTITY IfcContextDependentUnit
```

```
  SUBTYPE OF(IfcNamedUnit);
```

```
  Name : IfcLabel;
```

```
  INVERSE
```

```
  HasExternalReference : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;
```

```
END_ENTITY;
```

```
ENTITY IfcConversionBasedUnit
```

```
  SUBTYPE OF(IfcNamedUnit);
```

```
  Name : IfcLabel;
```

```
  ConversionFactor : IfcMeasureWithUnit;
```

```
  INVERSE
```

```
  HasExternalReference : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;
```

```
END_ENTITY;
```

```
ENTITY IfcConversionBasedUnitWithOffset
```

```
  SUBTYPE OF(IfcConversionBasedUnit);
```

```
  ConversionOffset : IfcReal;
```

```
END_ENTITY;
```

```
ENTITY IfcSIUnit
```

```
  SUBTYPE OF(IfcNamedUnit);
```

```
  Prefix : OPTIONAL IfcSIPrefix;
```

```
  Name : IfcSIUnitName;
```

```
  DERIVE
```

```
  SELF.IfNamedUnit.Dimensions : IfcDimensionalExponents := IfcDimensionsForSiUnit (SELF.Name);
```

```
END_ENTITY;
```

ENTITY IfcDimensionalExponents;

LengthExponent : INTEGER;
MassExponent : INTEGER;
TimeExponent : INTEGER;
ElectricCurrentExponent : INTEGER;
ThermodynamicTemperatureExponent : INTEGER;
AmountOfSubstanceExponent : INTEGER;
LuminousIntensityExponent : INTEGER;

END_ENTITY;

ENTITY IfcMonetaryUnit;

Currency : IfcLabel;

END_ENTITY;

ENTITY IfcReference;

TypeIdentifier : OPTIONAL IfcIdentifier;
AttributeIdentifier : OPTIONAL IfcIdentifier;
InstanceName : OPTIONAL IfcLabel;
ListPositions : OPTIONAL LIST [1:?] OF IfcInteger;
InnerReference : OPTIONAL IfcReference;

END_ENTITY;

ENTITY IfcPhysicalQuantity

ABSTRACT SUPERTYPE OF (ONEOF(IfcPhysicalComplexQuantity, IfcPhysicalSimpleQuantity));

Name : IfcLabel;

Description : OPTIONAL IfcText;

INVERSE

HasExternalReferences : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;

PartOfComplex : SET [0:1] OF IfcPhysicalComplexQuantity FOR HasQuantities;

END_ENTITY;

ENTITY IfcPhysicalComplexQuantity

SUBTYPE OF(IfcPhysicalQuantity);

HasQuantities : SET [1:?] OF IfcPhysicalQuantity;

Discrimination : IfcLabel;

Quality : OPTIONAL IfcLabel;

Usage : OPTIONAL IfcLabel;

WHERE

NOSELFREFERENCE : SIZEOF(QUERY(temp <* HasQuantities | SELF :=: temp)) = 0;

UNIQUEQUANTITYNAMES : IfcUniqueQuantityNames(HasQuantities);

END_ENTITY;

ENTITY IfcPhysicalSimpleQuantity

ABSTRACT SUPERTYPE OF (ONEOF(IfcQuantityArea, IfcQuantityCount, IfcQuantityLength,
IfcQuantityTime, IfcQuantityVolume, IfcQuantityWeight))

SUBTYPE OF(IfcPhysicalQuantity);

Unit : OPTIONAL IfcNamedUnit;

END_ENTITY;

ENTITY IfcQuantityArea

SUBTYPE OF(IfcPhysicalSimpleQuantity);

AreaValue : IfcAreaMeasure;

Formula : OPTIONAL IfcLabel;

WHERE

WR21 : NOT(EXISTS(SELFWIfcPhysicalSimpleQuantity.Unit)) OR

(SELFWIfcPhysicalSimpleQuantity.Unit.UnitType = IfcUnitEnum.AREAUNIT);

WR22 : AreaValue >= 0.;

END_ENTITY;

```
ENTITY IfcQuantityCount
  SUBTYPE OF(IfcPhysicalSimpleQuantity);
  CountValue : IfcCountMeasure;
  Formula    : OPTIONAL IfcLabel;
  WHERE
    WR21 : CountValue >= 0.;
END_ENTITY;
```

```
ENTITY IfcQuantityLength
  SUBTYPE OF(IfcPhysicalSimpleQuantity);
  LengthValue : IfcLengthMeasure;
  Formula     : OPTIONAL IfcLabel;
  WHERE
    WR21 : NOT(EXISTS(SELFWIfcPhysicalSimpleQuantity.Unit)) OR
          (SELFWIfcPhysicalSimpleQuantity.Unit.UnitType = IfcUnitEnum.LENGTHUNIT);
    WR22 : LengthValue >= 0.;
END_ENTITY;
```

```
ENTITY IfcQuantityTime
  SUBTYPE OF(IfcPhysicalSimpleQuantity);
  TimeValue : IfcTimeMeasure;
  Formula   : OPTIONAL IfcLabel;
  WHERE
    WR21 : NOT(EXISTS(SELFWIfcPhysicalSimpleQuantity.Unit)) OR
          (SELFWIfcPhysicalSimpleQuantity.Unit.UnitType = IfcUnitEnum.TIMEUNIT);
    WR22 : TimeValue >= 0.;
END_ENTITY;
```

```
ENTITY IfcQuantityVolume
  SUBTYPE OF(IfcPhysicalSimpleQuantity);
```

VolumeValue : IfcVolumeMeasure;

Formula : OPTIONAL IfcLabel;

WHERE

WR21 : NOT(EXISTS(SELFWIfcPhysicalSimpleQuantity.Unit)) OR

(SELFWIfcPhysicalSimpleQuantity.Unit.UnitType = IfcUnitEnum.VOLUMEUNIT);

WR22 : VolumeValue >= 0.;

END_ENTITY;

ENTITY IfcQuantityWeight

SUBTYPE OF(IfcPhysicalSimpleQuantity);

WeightValue : IfcMassMeasure;

Formula : OPTIONAL IfcLabel;

WHERE

WR21 : NOT(EXISTS(SELFWIfcPhysicalSimpleQuantity.Unit)) OR

(SELFWIfcPhysicalSimpleQuantity.Unit.UnitType = IfcUnitEnum.MASSUNIT);

WR22 : WeightValue >= 0.;

END_ENTITY;

ENTITY IfcCostSchedule

SUBTYPE OF(IfcControl);

PredefinedType : OPTIONAL IfcCostScheduleTypeEnum;

Status : OPTIONAL IfcLabel;

SubmittedOn : OPTIONAL IfcDateTime;

UpdateDate : OPTIONAL IfcDateTime;

END_ENTITY;

ENTITY IfcPerformanceHistory

SUBTYPE OF(IfcControl);

LifeCyclePhase : IfcLabel;

PredefinedType : OPTIONAL IfcPerformanceHistoryTypeEnum;

END_ENTITY;

ENTITY IfcPermit

SUBTYPE OF(IfcControl);

PredefinedType : OPTIONAL IfcPermitTypeEnum;

Status : OPTIONAL IfcLabel;

LongDescription : OPTIONAL IfcText;

END_ENTITY;

ENTITY IfcProjectOrder

SUBTYPE OF(IfcControl);

PredefinedType : OPTIONAL IfcProjectOrderTypeEnum;

Status : OPTIONAL IfcLabel;

LongDescription : OPTIONAL IfcText;

END_ENTITY;

ENTITY IfcWorkCalendar

SUBTYPE OF(IfcControl);

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

ExceptionTimes : OPTIONAL SET [1:?] OF IfcWorkTime;

PredefinedType : OPTIONAL IfcWorkCalendarTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcWorkCalendarTypeEnum.USERDEFINED) OR
((PredefinedType = IfcWorkCalendarTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcObject.ObjectType));

END_ENTITY;

ENTITY IfcWorkTime

SUBTYPE OF(IfcSchedulingTime);

```
    RecurrencePattern : OPTIONAL IfcRecurrencePattern;
    Start              : OPTIONAL IfcDate;
    Finish             : OPTIONAL IfcDate;
END_ENTITY;
```

```
ENTITY IfcSchedulingTime
    ABSTRACT SUPERTYPE OF (ONEOF(IfcEventTime, IfcLagTime, IfcResourceTime, IfcTaskTime,
IfcWorkTime));
    Name              : OPTIONAL IfcLabel;
    DataOrigin        : OPTIONAL IfcDataOriginEnum;
    UserDefinedDataOrigin : OPTIONAL IfcLabel;
END_ENTITY;
```

```
ENTITY IfcEventTime
    SUBTYPE OF(IfcSchedulingTime);
    ActualDate       : OPTIONAL IfcDateTime;
    EarlyDate        : OPTIONAL IfcDateTime;
    LateDate         : OPTIONAL IfcDateTime;
    ScheduleDate     : OPTIONAL IfcDateTime;
END_ENTITY;
```

```
ENTITY IfcLagTime
    SUBTYPE OF(IfcSchedulingTime);
    LagValue         : IfcTimeOrRatioSelect;
    DurationType     : IfcTaskDurationEnum;
END_ENTITY;
```

```
ENTITY IfcResourceTime
    SUBTYPE OF(IfcSchedulingTime);
    ScheduleWork     : OPTIONAL IfcDuration;
```

ScheduleUsage : OPTIONAL IfcPositiveRatioMeasure;
ScheduleStart : OPTIONAL IfcDateTime;
ScheduleFinish : OPTIONAL IfcDateTime;
ScheduleContour : OPTIONAL IfcLabel;
LevelingDelay : OPTIONAL IfcDuration;
IsOverAllocated : OPTIONAL IfcBoolean;
StatusTime : OPTIONAL IfcDateTime;
ActualWork : OPTIONAL IfcDuration;
ActualUsage : OPTIONAL IfcPositiveRatioMeasure;
ActualStart : OPTIONAL IfcDateTime;
ActualFinish : OPTIONAL IfcDateTime;
RemainingWork : OPTIONAL IfcDuration;
RemainingUsage : OPTIONAL IfcPositiveRatioMeasure;
Completion : OPTIONAL IfcPositiveRatioMeasure;

END_ENTITY;

ENTITY IfcTaskTime

SUBTYPE OF (IfcSchedulingTime);

DurationType : OPTIONAL IfcTaskDurationEnum;
ScheduleDuration : OPTIONAL IfcDuration;
ScheduleStart : OPTIONAL IfcDateTime;
ScheduleFinish : OPTIONAL IfcDateTime;
EarlyStart : OPTIONAL IfcDateTime;
EarlyFinish : OPTIONAL IfcDateTime;
LateStart : OPTIONAL IfcDateTime;
LateFinish : OPTIONAL IfcDateTime;
FreeFloat : OPTIONAL IfcDuration;
TotalFloat : OPTIONAL IfcDuration;
IsCritical : OPTIONAL IfcBoolean;
StatusTime : OPTIONAL IfcDateTime;

ActualDuration : OPTIONAL IfcDuration;
ActualStart : OPTIONAL IfcDateTime;
ActualFinish : OPTIONAL IfcDateTime;
RemainingTime : OPTIONAL IfcDuration;
Completion : OPTIONAL IfcPositiveRatioMeasure;

END_ENTITY;

ENTITY IfcTaskTimeRecurring

SUBTYPE OF(IfcTaskTime);

Recurrence : IfcRecurrencePattern;

END_ENTITY;

ENTITY IfcRecurrencePattern;

RecurrenceType : IfcRecurrenceTypeEnum;

DayComponent : OPTIONAL SET [1:?] OF IfcDayInMonthNumber;

WeekdayComponent : OPTIONAL SET [1:?] OF IfcDayInWeekNumber;

MonthComponent : OPTIONAL SET [1:?] OF IfcMonthInYearNumber;

Position : OPTIONAL IfcInteger;

Interval : OPTIONAL IfcInteger;

Occurrences : OPTIONAL IfcInteger;

TimePeriods : OPTIONAL LIST [1:?] OF IfcTimePeriod;

END_ENTITY;

ENTITY IfcTimePeriod;

StartTime : IfcTime;

EndTime : IfcTime;

END_ENTITY;

ENTITY IfcWorkControl

ABSTRACT SUPERTYPE OF (ONEOF(IfcWorkPlan, IfcWorkSchedule))

SUBTYPE OF(IfcControl);

CreationDate : IfcDateTime;

Creators : OPTIONAL SET [1:?] OF IfcPerson;

Purpose : OPTIONAL IfcLabel;

Duration : OPTIONAL IfcDuration;

TotalFloat : OPTIONAL IfcDuration;

StartTime : IfcDateTime;

FinishTime : OPTIONAL IfcDateTime;

END_ENTITY;

ENTITY IfcWorkPlan

SUBTYPE OF(IfcWorkControl);

PredefinedType : OPTIONAL IfcWorkPlanTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcWorkPlanTypeEnum.USERDEFINED) OR
EXISTS(SELFIfcObject.ObjectType);
((PredefinedType = IfcWorkPlanTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcObject.ObjectType));

END_ENTITY;

ENTITY IfcWorkSchedule

SUBTYPE OF(IfcWorkControl);

PredefinedType : OPTIONAL IfcWorkScheduleTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcWorkScheduleTypeEnum.USERDEFINED) OR
EXISTS(SELFIfcObject.ObjectType);
((PredefinedType = IfcWorkScheduleTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcObject.ObjectType));

END_ENTITY;

ENTITY IfcRelAssignsToControl

```

SUBTYPE OF(IfcRelAssigns);
    RelatingControl : IfcControl;
WHERE
    NOSELFREFERENCE : SIZEOF(QUERY(Temp < * SELFWIfcRelAssigns.RelatedObjects |
RelatingControl :=: Temp)) = 0;
END_ENTITY;

```

```

ENTITY IfcGroup
    SUPERTYPE OF (ONEOF(IfcAsset, IfcInventory, IfcStructuralLoadGroup, IfcStructuralResultGroup,
IfcSystem))
    SUBTYPE OF(IfcObject);
    INVERSE
        IsGroupedBy : SET OF IfcRelAssignsToGroup FOR RelatingGroup;
END_ENTITY;

```

```

ENTITY IfcAsset
    SUBTYPE OF(IfcGroup);
        Identification : OPTIONAL IfcIdentifier;
        OriginalValue : OPTIONAL IfcCostValue;
        CurrentValue : OPTIONAL IfcCostValue;
        TotalReplacementCost : OPTIONAL IfcCostValue;
        Owner : OPTIONAL IfcActorSelect;
        User : OPTIONAL IfcActorSelect;
        ResponsiblePerson : OPTIONAL IfcPerson;
        IncorporationDate : OPTIONAL IfcDate;
        DepreciatedValue : OPTIONAL IfcCostValue;
END_ENTITY;

```

```

ENTITY IfcInventory
    SUBTYPE OF(IfcGroup);

```

PredefinedType : OPTIONAL IfcInventoryTypeEnum;
Jurisdiction : OPTIONAL IfcActorSelect;
ResponsiblePersons : OPTIONAL SET [1:?] OF IfcPerson;
LastUpdateDate : OPTIONAL IfcDate;
CurrentValue : OPTIONAL IfcCostValue;
OriginalValue : OPTIONAL IfcCostValue;

END_ENTITY;

ENTITY IfcStructuralLoadGroup

SUBTYPE OF(IfcGroup);

PredefinedType : IfcLoadGroupTypeEnum;
ActionType : IfcActionTypeEnum;
ActionSource : IfcActionSourceTypeEnum;
Coefficient : OPTIONAL IfcRatioMeasure;
Purpose : OPTIONAL IfcLabel;

INVERSE

SourceOfResultGroup : SET [0:1] OF IfcStructuralResultGroup FOR ResultForLoadGroup;
LoadGroupFor : SET OF IfcStructuralAnalysisModel FOR LoadedBy;

WHERE

HASOBJECTTYPE : (
 (PredefinedType <> IfcLoadGroupTypeEnum.USERDEFINED) AND
 (ActionType <> IfcActionTypeEnum.USERDEFINED) AND
 (ActionSource <> IfcActionSourceTypeEnum.USERDEFINED)
) OR EXISTS(SELFWIfcObject.ObjectType);

END_ENTITY;

ENTITY IfcStructuralLoadCase

SUBTYPE OF(IfcStructuralLoadGroup);

SelfWeightCoefficients : OPTIONAL LIST [3:3] OF IfcRatioMeasure;

WHERE

```

        ISLOADCASEPREDEFINEDTYPE          :          SELFIfcStructuralLoadGroup.PredefinedType          =
IfcLoadGroupTypeEnum.LOAD_CASE;
END_ENTITY;

```

```

ENTITY IfcStructuralResultGroup

```

```

    SUBTYPE OF(IfcGroup);

```

```

        TheoryType          : IfcAnalysisTheoryTypeEnum;

```

```

        ResultForLoadGroup : OPTIONAL IfcStructuralLoadGroup;

```

```

        IsLinear           : IfcBoolean;

```

```

    INVERSE

```

```

        ResultGroupFor      : SET [0:1] OF IfcStructuralAnalysisModel FOR HasResults;

```

```

    WHERE

```

```

        HASOBJECTTYPE      :      (TheoryType      <>      IfcAnalysisTheoryTypeEnum.USERDEFINED)      OR
EXISTS(SELFWIfcObject.ObjectType);

```

```

END_ENTITY;

```

```

ENTITY IfcStructuralAnalysisModel

```

```

    SUBTYPE OF(IfcSystem);

```

```

        PredefinedType      : IfcAnalysisModelTypeEnum;

```

```

        OrientationOf2DPlane : OPTIONAL IfcAxis2Placement3D;

```

```

        LoadedBy            : OPTIONAL SET [1:?] OF IfcStructuralLoadGroup;

```

```

        HasResults          : OPTIONAL SET [1:?] OF IfcStructuralResultGroup;

```

```

        SharedPlacement     : OPTIONAL IfcObjectPlacement;

```

```

    WHERE

```

```

        HASOBJECTTYPE      :      (PredefinedType      <>      IfcAnalysisModelTypeEnum.USERDEFINED)      OR
EXISTS(SELFWIfcObject.ObjectType);

```

```

END_ENTITY;

```

```

ENTITY IfcSystem

```

```

    SUPERTYPE OF (ONEOF(IfcBuildingSystem, IfcDistributionSystem, IfcStructuralAnalysisModel,
IfcZone))

```



```

SUBTYPE OF(IfcGroup);
INVERSE
    ServicesBuildings : SET [0:1] OF IfcRelServicesBuildings FOR RelatingSystem;
END_ENTITY;

```

```

ENTITY IfcBuildingSystem
SUBTYPE OF(IfcSystem);
    PredefinedType : OPTIONAL IfcBuildingSystemTypeEnum;
    LongName       : OPTIONAL IfcLabel;
END_ENTITY;

```

```

ENTITY IfcDistributionSystem
SUBTYPE OF(IfcSystem);
    LongName       : OPTIONAL IfcLabel;
    PredefinedType : OPTIONAL IfcDistributionSystemEnum;
END_ENTITY;

```

```

ENTITY IfcDistributionCircuit
SUBTYPE OF(IfcDistributionSystem);
END_ENTITY;

```

```

ENTITY IfcZone
SUBTYPE OF(IfcSystem);
    LongName : OPTIONAL IfcLabel;
WHERE
    WR1 : (SIZEOF(SELFWIfcGroup.IsGroupedBy) = 0) OR
          (SIZEOF (QUERY (temp <* SELFWIfcGroup.IsGroupedBy[1].RelatedObjects |
NOT(('IFC4.IFCZONE' IN TYPEOF(temp)) OR
('IFC4.IFCSPACE' IN TYPEOF(temp)) OR
('IFC4.IFCSPATIALZONE' IN TYPEOF(temp))

```

```
))) = 0);  
END_ENTITY;
```

```
ENTITY IfcRelServicesBuildings  
  SUBTYPE OF(IfcRelConnects);  
  RelatingSystem : IfcSystem;  
  RelatedBuildings : SET [1:?] OF IfcSpatialElement;  
END_ENTITY;
```

```
ENTITY IfcRelConnects  
  ABSTRACT SUPERTYPE OF (ONEOF(IfcRelConnectsElements, IfcRelConnectsPortToElement,  
IfcRelConnectsPorts, IfcRelConnectsStructuralActivity, IfcRelConnectsStructuralMember,  
IfcRelContainedInSpatialStructure, IfcRelCoversBldgElements, IfcRelCoversSpaces,  
IfcRelFillsElement, IfcRelFlowControlElements, IfcRelInterferesElements,  
IfcRelReferencedInSpatialStructure, IfcRelSequence, IfcRelServicesBuildings, IfcRelSpaceBoundary))  
  SUBTYPE OF(IfcRelationship);  
END_ENTITY;
```

```
ENTITY IfcRelConnectsElements  
  SUPERTYPE OF (ONEOF(IfcRelConnectsPathElements, IfcRelConnectsWithRealizingElements))  
  SUBTYPE OF(IfcRelConnects);  
  ConnectionGeometry : OPTIONAL IfcConnectionGeometry;  
  RelatingElement : IfcElement;  
  RelatedElement : IfcElement;  
  WHERE  
    NOSELFREFERENCE : RelatingElement :<>: RelatedElement;  
END_ENTITY;
```

```
ENTITY IfcRelConnectsPathElements  
  SUBTYPE OF(IfcRelConnectsElements);  
  RelatingPriorities : LIST OF IfcInteger;
```

RelatedPriorities : LIST OF IfcInteger;
RelatedConnectionType : IfcConnectionTypeEnum;
RelatingConnectionType : IfcConnectionTypeEnum;

WHERE

NORMALIZEDRELATINGPRIORITIES : (SIZEOF(RelatingPriorities) = 0)
OR
(SIZEOF (QUERY (temp <* RelatingPriorities
| {0 <= temp <= 100}
)) = SIZEOF(RelatingPriorities));
NORMALIZEDRELATEDPRIORITIES : (SIZEOF(RelatedPriorities) = 0)
OR
(SIZEOF (QUERY (temp <* RelatedPriorities
| {0 <= temp <= 100}
)) = SIZEOF(RelatedPriorities));

END_ENTITY;

ENTITY IfcRelConnectsWithRealizingElements

SUBTYPE OF(IfcRelConnectsElements);
RealizingElements : SET [1:?] OF IfcElement;
ConnectionType : OPTIONAL IfcLabel;

END_ENTITY;

ENTITY IfcElement

ABSTRACT SUPERTYPE OF (ONEOF(IfcBuildingElement, IfcCivilElement, IfcDistributionElement,
IfcElementAssembly, IfcElementComponent, IfcFeatureElement, IfcFurnishingElement,
IfcGeographicElement, IfcTransportElement, IfcVirtualElement))

SUBTYPE OF(IfcProduct);
Tag : OPTIONAL IfcIdentifier;

INVERSE

FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;

```

ConnectedTo          : SET OF IfcRelConnectsElements FOR RelatingElement;
IsInterferedByElements : SET OF IfcRelInterferesElements FOR RelatedElement;
InterferesElements   : SET OF IfcRelInterferesElements FOR RelatingElement;
HasProjections        : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasOpenings           : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries    : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom         : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure  : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
HasCoverings          : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
END_ENTITY;

```

```

ENTITY IfcProduct

```

```

    ABSTRACT SUPERTYPE OF (ONEOF(IfcAnnotation, IfcElement, IfcGrid, IfcPort, IfcProxy,
IfcSpatialElement, IfcStructuralActivity, IfcStructuralItem, IfcPositioningElement))

```

```

    SUBTYPE OF(IfcObject);

```

```

        ObjectPlacement : OPTIONAL IfcObjectPlacement;

```

```

        Representation : OPTIONAL IfcProductRepresentation;

```

```

    INVERSE

```

```

        ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;

```

```

    WHERE

```

```

        PLACEMENTFORSHAPEREPRESENTATION : (EXISTS(Representation) AND EXISTS(ObjectPlacement))

```

```

            OR (EXISTS(Representation) AND

```

```

                (SIZEOF(QUERY(temp < * Representation.Representations |
'IFC4.IFCSHAPEREPRESENTATION' IN TYPEOF(temp))) = 0))

```

```

            OR (NOT(EXISTS(Representation)));

```

```

END_ENTITY;

```

```

ENTITY IfcAnnotation

```

```

SUBTYPE OF(IfcProduct);
INVERSE
    ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
END_ENTITY;

```

```

ENTITY IfcRelContainedInSpatialStructure
    SUBTYPE OF(IfcRelConnects);
    RelatedElements : SET [1:?] OF IfcProduct;
    RelatingStructure : IfcSpatialElement;
    WHERE
        WR31 : SIZEOF(QUERY(temp <* RelatedElements | 'IFC4.IFCSPATIALSTRUCTUREELEMENT' IN
        TYPEOF(temp))) = 0;
END_ENTITY;

```

```

ENTITY IfcSpatialElement
    ABSTRACT SUPERTYPE OF (ONEOF(IfcExternalSpatialStructureElement, IfcSpatialStructureElement,
    IfcSpatialZone, IfcCivilSpatialStructureElement_K, IfcCivilSpatialBoundary_K))
    SUBTYPE OF(IfcProduct);
    LongName : OPTIONAL IfcLabel;
    INVERSE
        ContainsElements : SET OF IfcRelContainedInSpatialStructure FOR RelatingStructure;
        ServicedBySystems : SET OF IfcRelServicesBuildings FOR RelatedBuildings;
        ReferencesElements : SET OF IfcRelReferencedInSpatialStructure FOR RelatingStructure;
END_ENTITY;

```

```

ENTITY IfcExternalSpatialStructureElement
    ABSTRACT SUPERTYPE
    SUBTYPE OF(IfcSpatialElement);
END_ENTITY;

```

```

ENTITY IfcExternalSpatialElement
  SUBTYPE OF(IfcExternalSpatialStructureElement);
  PredefinedType : OPTIONAL IfcExternalSpatialElementTypeEnum;
  INVERSE
  BoundedBy      : SET OF IfcRelSpaceBoundary FOR RelatingSpace;
END_ENTITY;

```

```

ENTITY IfcRelSpaceBoundary
  SUBTYPE OF(IfcRelConnects);
  RelatingSpace      : IfcSpaceBoundarySelect;
  RelatedBuildingElement : IfcElement;
  ConnectionGeometry : OPTIONAL IfcConnectionGeometry;
  PhysicalOrVirtualBoundary : IfcPhysicalOrVirtualEnum;
  InternalOrExternalBoundary : IfcInternalOrExternalEnum;
  WHERE
  CORRECTPHYSORVIRT : ((PhysicalOrVirtualBoundary = IfcPhysicalOrVirtualEnum.Physical)
    AND (NOT('IFC4.IFCVIRTUALELEMENT' IN TYPEOF(RelatedBuildingElement))))
    OR
    ((PhysicalOrVirtualBoundary = IfcPhysicalOrVirtualEnum.Virtual)
    AND (('IFC4.IFCVIRTUALELEMENT' IN TYPEOF(RelatedBuildingElement))
    OR ('IFC4.IFCOPENINGELEMENT' IN TYPEOF(RelatedBuildingElement))))
    OR
    (PhysicalOrVirtualBoundary = IfcPhysicalOrVirtualEnum.NotDefined);
END_ENTITY;

```

```

ENTITY IfcRelSpaceBoundary1stLevel
  SUBTYPE OF(IfcRelSpaceBoundary);
  ParentBoundary : OPTIONAL IfcRelSpaceBoundary1stLevel;
  INVERSE
  InnerBoundaries : SET OF IfcRelSpaceBoundary1stLevel FOR ParentBoundary;

```

END_ENTITY;

ENTITY IfcRelSpaceBoundary2ndLevel

SUBTYPE OF(IfcRelSpaceBoundary1stLevel);

CorrespondingBoundary : OPTIONAL IfcRelSpaceBoundary2ndLevel;

INVERSE

Corresponds : SET [0:1] OF IfcRelSpaceBoundary2ndLevel FOR CorrespondingBoundary;

END_ENTITY;

ENTITY IfcConnectionGeometry

ABSTRACT SUPERTYPE OF (ONEOF(IfcConnectionCurveGeometry, IfcConnectionPointGeometry, IfcConnectionSurfaceGeometry, IfcConnectionVolumeGeometry));

END_ENTITY;

ENTITY IfcConnectionCurveGeometry

SUBTYPE OF(IfcConnectionGeometry);

CurveOnRelatingElement : IfcCurveOrEdgeCurve;

CurveOnRelatedElement : OPTIONAL IfcCurveOrEdgeCurve;

END_ENTITY;

ENTITY IfcEdgeCurve

SUBTYPE OF(IfcEdge);

EdgeGeometry : IfcCurve;

SameSense : IfcBoolean;

END_ENTITY;

ENTITY IfcEdge

SUPERTYPE OF (ONEOF(IfcEdgeCurve, IfcOrientedEdge, IfcSubedge))

SUBTYPE OF(IfcTopologicalRepresentationItem);

EdgeStart : IfcVertex;

```
EdgeEnd : IfcVertex;  
END_ENTITY;
```

```
ENTITY IfcTopologicalRepresentationItem  
  ABSTRACT SUPERTYPE OF (ONEOF(IfcConnectedFaceSet, IfcEdge, IfcFace, IfcFaceBound, IfcLoop,  
IfcPath, IfcVertex))  
  SUBTYPE OF(IfcRepresentationItem);  
END_ENTITY;
```

```
ENTITY IfcConnectedFaceSet  
  SUPERTYPE OF (ONEOF(IfcClosedShell, IfcOpenShell))  
  SUBTYPE OF(IfcTopologicalRepresentationItem);  
  CfsFaces : SET [1:?] OF IfcFace;  
END_ENTITY;
```

```
ENTITY IfcClosedShell  
  SUBTYPE OF(IfcConnectedFaceSet);  
END_ENTITY;
```

```
ENTITY IfcOpenShell  
  SUBTYPE OF(IfcConnectedFaceSet);  
END_ENTITY;
```

```
ENTITY IfcFace  
  SUBTYPE OF(IfcTopologicalRepresentationItem);  
  Bounds : SET [1:?] OF IfcFaceBound;  
  INVERSE  
  HasTextureMaps : SET OF IfcTextureMap FOR MappedTo;  
  WHERE  
  HASOUTERBOUND : SIZEOF(QUERY(temp <* Bounds | 'IFC4.IFCFACEOUTERBOUND' IN TYPEOF(temp))) <=
```


1;

END_ENTITY;

ENTITY IfcFaceSurface

SUBTYPE OF(IfcFace);

FaceSurface : IfcSurface;

SameSense : IfcBoolean;

END_ENTITY;

ENTITY IfcAdvancedFace

SUBTYPE OF(IfcFaceSurface);

WHERE

APPLICABLESURFACE : SIZEOF (

['IFC4.IFCELEMENTARYSURFACE',

'IFC4.IFCSWEPTSURFACE',

'IFC4.IFCBSPLINESURFACE'] *

TYPEOF(SELFWIfcFaceSurface.FaceSurface)) = 1;

REQUIRESEDGECURVE : SIZEOF(QUERY (ElpFbnds <*
QUERY (Bnds <* SELFWIfcFace.Bounds |
'IFC4.IFCEDGELOOP' IN TYPEOF(Bnds.Bound)) |
NOT (SIZEOF (QUERY (Oe <* ElpFbnds.BoundWIfcEdgeLoop.EdgeList |
NOT('IFC4.IFCEDGECURVE' IN
TYPEOF(OeWIfcOrientedEdge.EdgeElement)
))) = 0
))) = 0;

APPLICABLEEDGECURVES : SIZEOF(QUERY (ElpFbnds <*
QUERY (Bnds <* SELFWIfcFace.Bounds |
'IFC4.IFCEDGELOOP' IN TYPEOF(Bnds.Bound)) |
NOT (SIZEOF (QUERY (Oe <* ElpFbnds.BoundWIfcEdgeLoop.EdgeList |
NOT (SIZEOF (['IFC4.IFCLINE',

```

        ' IFC4. IFCCONIC',
        ' IFC4. IFCPOLYLINE',
        ' IFC4. IFCBSPLINECURVE'] *
        TYPEOF(OneOf(IfcOrientedEdge, EdgeElementWithEdgeCurve, EdgeGeometry)) =
1 )
        )) = 0
        ))) = 0;

```

END_ENTITY;

ENTITY IfcFaceBound

SUBTYPE OF(IfcTopologicalRepresentationItem);

Bound : IfcLoop;

Orientation : IfcBoolean;

END_ENTITY;

ENTITY IfcFaceOuterBound

SUBTYPE OF(IfcFaceBound);

END_ENTITY;

ENTITY IfcLoop

SUPERTYPE OF (ONEOF(IfcEdgeLoop, IfcPolyLoop, IfcVertexLoop))

SUBTYPE OF(IfcTopologicalRepresentationItem);

END_ENTITY;

ENTITY IfcEdgeLoop

SUBTYPE OF(IfcLoop);

EdgeList : LIST [1:?] OF IfcOrientedEdge;

DERIVE

Ne : IfcInteger := SIZEOF(EdgeList);

WHERE

```

    ISCLOSED      : (EdgeList[1].EdgeStart) ::= (EdgeList[Ne].EdgeEnd);
    ISCONTINUOUS  : IfcLoopHeadToTail(SELF);
END_ENTITY;

ENTITY IfcOrientedEdge
  SUBTYPE OF(IfcEdge);
  EdgeElement      : IfcEdge;
  Orientation      : IfcBoolean;
  DERIVE
    SELFWIfcEdge.EdgeStart : IfcVertex := IfcBooleanChoose
                                (Orientation, EdgeElement.EdgeStart, EdgeElement.EdgeEnd);
    SELFWIfcEdge.EdgeEnd   : IfcVertex := IfcBooleanChoose
                                (Orientation, EdgeElement.EdgeEnd, EdgeElement.EdgeStart);
  WHERE
    EDGEELEMENTNOTORIENTED : NOT('IFC4.IFCORIENTEDEDGE' IN TYPEOF(EdgeElement));
END_ENTITY;

ENTITY IfcVertex
  SUBTYPE OF(IfcTopologicalRepresentationItem);
END_ENTITY;

ENTITY IfcVertexPoint
  SUBTYPE OF(IfcVertex);
  VertexGeometry : IfcPoint;
END_ENTITY;

ENTITY IfcPolyLoop
  SUBTYPE OF(IfcLoop);
  Polygon : LIST [3:?] OF IfcCartesianPoint;
  WHERE

```

```
    ALLPOINTSSAMEDIM : SIZEOF(QUERY(Temp <* Polygon | Temp.Dim <> Polygon[1].Dim)) = 0;
END_ENTITY;
```

```
ENTITY IfcVertexLoop
    SUBTYPE OF(IfcLoop);
    LoopVertex : IfcVertex;
END_ENTITY;
```

```
ENTITY IfcTextureMap
    SUBTYPE OF(IfcTextureCoordinate);
    Vertices : LIST [3:?] OF IfcTextureVertex;
    MappedTo : IfcFace;
END_ENTITY;
```

```
ENTITY IfcTextureCoordinate
    ABSTRACT SUPERTYPE OF (ONEOF(IfcIndexedTextureMap, IfcTextureCoordinateGenerator,
IfcTextureMap))
    SUBTYPE OF(IfcPresentationItem);
    Maps : LIST [1:?] OF IfcSurfaceTexture;
END_ENTITY;
```

```
ENTITY IfcPresentationItem
    ABSTRACT SUPERTYPE OF (ONEOF(IfcColourRgbList, IfcColourSpecification, IfcCurveStyleFont,
IfcCurveStyleFontAndScaling, IfcCurveStyleFontPattern, IfcIndexedColourMap, IfcPreDefinedItem,
IfcSurfaceStyleLighting, IfcSurfaceStyleRefraction, IfcSurfaceStyleShading,
IfcSurfaceStyleWithTextures, IfcSurfaceTexture, IfcTextStyleForDefinedFont, IfcTextStyleTextModel,
IfcTextureCoordinate, IfcTextureVertex, IfcTextureVertexList));
END_ENTITY;
```

```
ENTITY IfcColourRgbList
    SUBTYPE OF(IfcPresentationItem);
```

```
    ColourList : LIST [1:?] OF LIST [3:3] OF IfcNormalisedRatioMeasure;  
END_ENTITY;
```

```
ENTITY IfcColourSpecification  
  ABSTRACT SUPERTYPE  
  SUBTYPE OF(IfcPresentationItem);  
  Name : OPTIONAL IfcLabel;  
END_ENTITY;
```

```
ENTITY IfcColourRgb  
  SUBTYPE OF(IfcColourSpecification);  
  Red   : IfcNormalisedRatioMeasure;  
  Green : IfcNormalisedRatioMeasure;  
  Blue  : IfcNormalisedRatioMeasure;  
END_ENTITY;
```

```
ENTITY IfcCurveStyleFont  
  SUBTYPE OF(IfcPresentationItem);  
  Name      : OPTIONAL IfcLabel;  
  PatternList : LIST [1:?] OF IfcCurveStyleFontPattern;  
END_ENTITY;
```

```
ENTITY IfcCurveStyleFontPattern  
  SUBTYPE OF(IfcPresentationItem);  
  VisibleSegmentLength : IfcLengthMeasure;  
  InvisibleSegmentLength : IfcPositiveLengthMeasure;  
  WHERE  
  VISIBLELENGTHGREATEREQUALZERO : VisibleSegmentLength >= 0.;  
END_ENTITY;
```

```

ENTITY IfcCurveStyleFontAndScaling
  SUBTYPE OF(IfcPresentationItem);
  Name          : OPTIONAL IfcLabel;
  CurveFont     : IfcCurveStyleFontSelect;
  CurveFontScaling : IfcPositiveRatioMeasure;
END_ENTITY;

```

```

ENTITY IfcPreDefinedCurveFont
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcPreDefinedItem);
END_ENTITY;

```

```

ENTITY IfcPreDefinedItem
  ABSTRACT SUPERTYPE OF (ONEOF(IfcPreDefinedColour, IfcPreDefinedCurveFont,
IfcPreDefinedTextFont))
  SUBTYPE OF(IfcPresentationItem);
  Name : IfcLabel;
END_ENTITY;

```

```

ENTITY IfcPreDefinedColour
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcPreDefinedItem);
END_ENTITY;

```

```

ENTITY IfcDraftingPreDefinedColour
  SUBTYPE OF(IfcPreDefinedColour);
  WHERE
    PREDEFINEDCOLOURNAMES : SELFWIfcPreDefinedItem.Name IN
['black', 'red', 'green', 'blue', 'yellow',
'magenta', 'cyan', 'white', 'by layer'];

```

END_ENTITY;

ENTITY IfcPreDefinedTextFont

ABSTRACT SUPERTYPE

SUBTYPE OF(IfcPreDefinedItem);

END_ENTITY;

ENTITY IfcTextStyleFontModel

SUBTYPE OF(IfcPreDefinedTextFont);

FontFamily : LIST [1:?] OF IfcTextFontName;

FontStyle : OPTIONAL IfcFontStyle;

FontVariant : OPTIONAL IfcFontVariant;

FontWeight : OPTIONAL IfcFontWeight;

FontSize : IfcSizeSelect;

WHERE

MEASUREOFFONTSIZE : ('IFC4.IFCLENGTHMEASURE' IN TYPEOF(SELF.FontSize)) AND
(SELF.FontSize > 0.);

END_ENTITY;

ENTITY IfcDraftingPreDefinedCurveFont

SUBTYPE OF(IfcPreDefinedCurveFont);

WHERE

PREDEFINEDCURVEFONTNAMES : SELF.IfcpreddefinedItem.Name IN
['continuous',
'chain',
'chain double dash',
'dashed',
'dotted',
'by layer'];

END_ENTITY;

```

ENTITY IfcIndexedColourMap
  SUBTYPE OF(IfcPresentationItem);
  MappedTo      : IfcTessellatedFaceSet;
  Opacity       : OPTIONAL IfcNormalisedRatioMeasure;
  Colours       : IfcColourRgbList;
  ColourIndex  : LIST [1:?] OF IfcPositiveInteger;
END_ENTITY;

```

```

ENTITY IfcTessellatedFaceSet
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcTessellatedItem);
  Coordinates   : IfcCartesianPointList3D;
  Normals       : OPTIONAL LIST [1:?] OF LIST [3:3] OF IfcParameterValue;
  Closed        : OPTIONAL IfcBoolean;
  INVERSE
  HasColours    : SET [0:1] OF IfcIndexedColourMap FOR MappedTo;
  HasTextures   : SET OF IfcIndexedTextureMap FOR MappedTo;
END_ENTITY;

```

```

ENTITY IfcTessellatedItem
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcGeometricRepresentationItem);
END_ENTITY;

```

```

ENTITY IfcTriangulatedFaceSet
  SUBTYPE OF(IfcTessellatedFaceSet);
  CoordIndex    : LIST [1:?] OF LIST [3:3] OF IfcPositiveInteger;
  NormalIndex   : OPTIONAL LIST [1:?] OF LIST [3:3] OF IfcPositiveInteger;
  DERIVE

```



```
    NumberOfTriangles : IfcInteger := SIZEOF(CoordIndex);  
END_ENTITY;
```

```
ENTITY IfcIndexedTextureMap  
  ABSTRACT SUPERTYPE  
  SUBTYPE OF(IfcTextureCoordinate);  
  MappedTo : IfcTessellatedFaceSet;  
  TexCoords : IfcTextureVertexList;  
END_ENTITY;
```

```
ENTITY IfcIndexedTriangleTextureMap  
  SUBTYPE OF(IfcIndexedTextureMap);  
  TexCoordIndex : OPTIONAL LIST [1:?] OF LIST [3:3] OF IfcPositiveInteger;  
END_ENTITY;
```

```
ENTITY IfcTextureVertexList  
  SUBTYPE OF(IfcPresentationItem);  
  TexCoordsList : LIST [1:?] OF LIST [2:2] OF IfcParameterValue;  
END_ENTITY;
```

```
ENTITY IfcSurfaceStyleLighting  
  SUBTYPE OF(IfcPresentationItem);  
  DiffuseTransmissionColour : IfcColourRgb;  
  DiffuseReflectionColour   : IfcColourRgb;  
  TransmissionColour       : IfcColourRgb;  
  ReflectanceColour        : IfcColourRgb;  
END_ENTITY;
```

```
ENTITY IfcSurfaceStyleRefraction  
  SUBTYPE OF(IfcPresentationItem);
```

```
    RefractionIndex : OPTIONAL IfcReal;  
    DispersionFactor : OPTIONAL IfcReal;  
END_ENTITY;
```

```
ENTITY IfcSurfaceStyleShading  
    SUBTYPE OF(IfcPresentationItem);  
    SurfaceColour : IfcColourRgb;  
    Transparency : OPTIONAL IfcNormalisedRatioMeasure;  
END_ENTITY;
```

```
ENTITY IfcSurfaceStyleRendering  
    SUBTYPE OF(IfcSurfaceStyleShading);  
    DiffuseColour : OPTIONAL IfcColourOrFactor;  
    TransmissionColour : OPTIONAL IfcColourOrFactor;  
    DiffuseTransmissionColour : OPTIONAL IfcColourOrFactor;  
    ReflectionColour : OPTIONAL IfcColourOrFactor;  
    SpecularColour : OPTIONAL IfcColourOrFactor;  
    SpecularHighlight : OPTIONAL IfcSpecularHighlightSelect;  
    ReflectanceMethod : IfcReflectanceMethodEnum;  
END_ENTITY;
```

```
ENTITY IfcSurfaceStyleWithTextures  
    SUBTYPE OF(IfcPresentationItem);  
    Textures : LIST [1:?] OF IfcSurfaceTexture;  
END_ENTITY;
```

```
ENTITY IfcSurfaceTexture  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcBlobTexture, IfcImageTexture, IfcPixelTexture))  
    SUBTYPE OF(IfcPresentationItem);  
    RepeatS : IfcBoolean;
```

```

RepeatT      : IfcBoolean;
Mode        : OPTIONAL IfcIdentifier;
TextureTransform : OPTIONAL IfcCartesianTransformationOperator2D;
Parameter    : OPTIONAL LIST [1:?] OF IfcIdentifier;
INVERSE
  IsMappedBy      : SET OF IfcTextureCoordinate FOR Maps;
  UsedInStyles    : SET OF IfcSurfaceStyleWithTextures FOR Textures;
END_ENTITY;

```

```

ENTITY IfcBlobTexture
  SUBTYPE OF(IfcSurfaceTexture);
  RasterFormat : IfcIdentifier;
  RasterCode   : IfcBinary;
  WHERE
    SUPPORTEDRASTERFORMAT : SELF.RasterFormat IN ['BMP', 'JPG', 'GIF', 'PNG'];
    RASTERCODEBYTESTREAM  : BLENGTH(RasterCode) MOD 8 = 0;
END_ENTITY;

```

```

ENTITY IfcImageTexture
  SUBTYPE OF(IfcSurfaceTexture);
  URLReference : IfcURIReference;
END_ENTITY;

```

```

ENTITY IfcPixelTexture
  SUBTYPE OF(IfcSurfaceTexture);
  Width      : IfcInteger;
  Height     : IfcInteger;
  ColourComponents : IfcInteger;
  Pixel      : LIST [1:?] OF IfcBinary;
  WHERE

```

```

MINPIXELINS           : Width >= 1;
MINPIXELINT           : Height >= 1;
NUMBEROFCOLOURS      : {1 <= ColourComponents <= 4};
SIZEOFPIXELLIST       : SIZEOF(Pixel) = (Width * Height);
PIXELASBYTEANDSAMELENGTH : SIZEOF(QUERY(temp<* Pixel |
                                (BLENGTH(temp) MOD 8 = 0) AND
                                (BLENGTH(temp) = BLENGTH(Pixel[1]))
                                )) = SIZEOF(Pixel);

```

END_ENTITY;

ENTITY IfcTextStyleForDefinedFont

SUBTYPE OF(IfcPresentationItem);

Colour : IfcColour;

BackgroundColour : OPTIONAL IfcColour;

END_ENTITY;

ENTITY IfcTextStyleTextModel

SUBTYPE OF(IfcPresentationItem);

TextIndent : OPTIONAL IfcSizeSelect;

TextAlign : OPTIONAL IfcTextAlignment;

TextDecoration : OPTIONAL IfcTextDecoration;

LetterSpacing : OPTIONAL IfcSizeSelect;

WordSpacing : OPTIONAL IfcSizeSelect;

TextTransform : OPTIONAL IfcTextTransformation;

LineHeight : OPTIONAL IfcSizeSelect;

END_ENTITY;

ENTITY IfcTextureVertex

SUBTYPE OF(IfcPresentationItem);

Coordinates : LIST [2:2] OF IfcParameterValue;

END_ENTITY;

ENTITY IfcTextureCoordinateGenerator

SUBTYPE OF(IfcTextureCoordinate);

Mode : IfcLabel;

Parameter : OPTIONAL LIST [1:?] OF IfcReal;

END_ENTITY;

ENTITY IfcPath

SUBTYPE OF(IfcTopologicalRepresentationItem);

EdgeList : LIST [1:?] OF IfcOrientedEdge;

WHERE

ISCONTINUOUS : IfcPathHeadToTail(SELF);

END_ENTITY;

ENTITY IfcSubedge

SUBTYPE OF(IfcEdge);

ParentEdge : IfcEdge;

END_ENTITY;

ENTITY IfcConnectionPointGeometry

SUBTYPE OF(IfcConnectionGeometry);

PointOnRelatingElement : IfcPointOrVertexPoint;

PointOnRelatedElement : OPTIONAL IfcPointOrVertexPoint;

END_ENTITY;

ENTITY IfcConnectionPointEccentricity

SUBTYPE OF(IfcConnectionPointGeometry);

EccentricityInX : OPTIONAL IfcLengthMeasure;

EccentricityInY : OPTIONAL IfcLengthMeasure;

```
    EccentricityInZ : OPTIONAL IfcLengthMeasure;  
END_ENTITY;
```

```
ENTITY IfcConnectionSurfaceGeometry  
    SUBTYPE OF(IfcConnectionGeometry);  
    SurfaceOnRelatingElement : IfcSurfaceOrFaceSurface;  
    SurfaceOnRelatedElement : OPTIONAL IfcSurfaceOrFaceSurface;  
END_ENTITY;
```

```
ENTITY IfcFaceBasedSurfaceModel  
    SUBTYPE OF(IfcGeometricRepresentationItem);  
    FbsmFaces : SET [1:?] OF IfcConnectedFaceSet;  
    DERIVE  
        Dim : IfcDimensionCount := 3;  
END_ENTITY;
```

```
ENTITY IfcConnectionVolumeGeometry  
    SUBTYPE OF(IfcConnectionGeometry);  
    VolumeOnRelatingElement : IfcSolidOrShell;  
    VolumeOnRelatedElement : OPTIONAL IfcSolidOrShell;  
END_ENTITY;
```

```
ENTITY IfcSolidModel  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcCsgSolid, IfcManifoldSolidBrep, IfcSweptAreaSolid,  
IfcSweptDiskSolid))  
    SUBTYPE OF(IfcGeometricRepresentationItem);  
    DERIVE  
        Dim : IfcDimensionCount := 3;  
END_ENTITY;
```

```

ENTITY IfcCsgSolid
  SUBTYPE OF(IfcSolidModel);
  TreeRootExpression : IfcCsgSelect;
END_ENTITY;

```

```

ENTITY IfcBooleanResult
  SUBTYPE OF(IfcGeometricRepresentationItem);
  Operator      : IfcBooleanOperator;
  FirstOperand  : IfcBooleanOperand;
  SecondOperand : IfcBooleanOperand;
  DERIVE
  Dim           : IfcDimensionCount := FirstOperand.Dim;
  WHERE
  SAMEDIM : FirstOperand.Dim = SecondOperand.Dim;
END_ENTITY;

```

```

ENTITY IfcBooleanClippingResult
  SUBTYPE OF(IfcBooleanResult);
  WHERE
  FIRSTOPERANDTYPE : ('IFC4.IFCSWEPTAREASOLID' IN TYPEOF(FirstOperand)) OR
                    ('IFC4.IFCSWEPTDISCSOLID' IN TYPEOF(FirstOperand)) OR
                    ('IFC4.IFCBOOLEANCLIPPINGRESULT' IN TYPEOF(FirstOperand));
  SECONDOPERANDTYPE : ('IFC4.IFCHALFSPACESOLID' IN TYPEOF(SecondOperand));
  OPERATORATYPE     : Operator = DIFFERENCE;
END_ENTITY;

```

```

ENTITY IfcCsgPrimitive3D
  ABSTRACT SUPERTYPE OF (ONEOF(IfcBlock, IfcRectangularPyramid, IfcRightCircularCone,
IfcRightCircularCylinder, IfcSphere))
  SUBTYPE OF(IfcGeometricRepresentationItem);

```

```
    Position : IfcAxis2Placement3D;  
DERIVE  
    Dim      : IfcDimensionCount := 3;  
END_ENTITY;
```

```
ENTITY IfcBlock  
    SUBTYPE OF(IfcCsgPrimitive3D);  
    XLength : IfcPositiveLengthMeasure;  
    YLength : IfcPositiveLengthMeasure;  
    ZLength : IfcPositiveLengthMeasure;  
END_ENTITY;
```

```
ENTITY IfcRectangularPyramid  
    SUBTYPE OF(IfcCsgPrimitive3D);  
    XLength : IfcPositiveLengthMeasure;  
    YLength : IfcPositiveLengthMeasure;  
    Height  : IfcPositiveLengthMeasure;  
END_ENTITY;
```

```
ENTITY IfcRightCircularCone  
    SUBTYPE OF(IfcCsgPrimitive3D);  
    Height      : IfcPositiveLengthMeasure;  
    BottomRadius : IfcPositiveLengthMeasure;  
END_ENTITY;
```

```
ENTITY IfcRightCircularCylinder  
    SUBTYPE OF(IfcCsgPrimitive3D);  
    Height : IfcPositiveLengthMeasure;  
    Radius : IfcPositiveLengthMeasure;  
END_ENTITY;
```



```

ENTITY IfcSphere
  SUBTYPE OF(IfcCsgPrimitive3D);
  Radius : IfcPositiveLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcHalfSpaceSolid
  SUPERTYPE OF (ONEOF(IfcBoxedHalfSpace, IfcPolygonalBoundedHalfSpace))
  SUBTYPE OF(IfcGeometricRepresentationItem);
  BaseSurface : IfcSurface;
  AgreementFlag : IfcBoolean;
  DERIVE
  Dim : IfcDimensionCount := 3;
END_ENTITY;

```

```

ENTITY IfcBoxedHalfSpace
  SUBTYPE OF(IfcHalfSpaceSolid);
  Enclosure : IfcBoundingBox;
  WHERE
  UNBOUNDEDSURFACE : NOT ('IFC4.IFCURVEBOUNDEDPLANE' IN
  TYPEOF(SELF|IfcHalfSpaceSolid.BaseSurface));
END_ENTITY;

```

```

ENTITY IfcBoundingBox
  SUBTYPE OF(IfcGeometricRepresentationItem);
  Corner : IfcCartesianPoint;
  XDim : IfcPositiveLengthMeasure;
  YDim : IfcPositiveLengthMeasure;
  ZDim : IfcPositiveLengthMeasure;
  DERIVE

```

```
    Dim      : IfcDimensionCount := 3;
END_ENTITY;
```

```
ENTITY IfcPolygonalBoundedHalfSpace
```

```
  SUBTYPE OF(IfcHalfSpaceSolid);
```

```
    Position      : IfcAxis2Placement3D;
```

```
    PolygonalBoundary : IfcBoundedCurve;
```

```
  WHERE
```

```
    BOUNDARYDIM : PolygonalBoundary.Dim = 2;
```

```
    BOUNDARYTYPE : SIZEOF(TYPEOF(PolygonalBoundary) * [
      'IFC4.IFCPOLYLINE',
      'IFC4.IFCCOMPOSITECURVE' ]
    ) = 1;
```

```
END_ENTITY;
```

```
ENTITY IfcManifoldSolidBrep
```

```
  ABSTRACT SUPERTYPE OF (ONEOF(IfcAdvancedBrep, IfcFacetedBrep))
```

```
  SUBTYPE OF(IfcSolidModel);
```

```
    Outer : IfcClosedShell;
```

```
END_ENTITY;
```

```
ENTITY IfcAdvancedBrep
```

```
  SUBTYPE OF(IfcManifoldSolidBrep);
```

```
  WHERE
```

```
    HASADVANCEDFACES : SIZEOF(QUERY(Afs <* SELFIfcManifoldSolidBrep.Outer.CfsFaces |
      (NOT ('IFC4.IFCADVANCEDFACE' IN TYPEOF(Afs)))
    )) = 0;
```

```
END_ENTITY;
```

```
ENTITY IfcAdvancedBrepWithVoids
```

SUBTYPE OF(IfcAdvancedBrep);

Voids : SET [1:?] OF IfcClosedShell;

WHERE

VOIDSHAVEADVANCEDFACES : SIZEOF (QUERY (Vsh <* Voids |
SIZEOF (QUERY (Afs <* Vsh.CfsFaces |
(NOT ('IFC4.IFCADVANCEDFACE' IN TYPEOF(Afs)))
)) = 0
)) = 0;

END_ENTITY;

ENTITY IfcFacetedBrep

SUBTYPE OF(IfcManifoldSolidBrep);

END_ENTITY;

ENTITY IfcFacetedBrepWithVoids

SUBTYPE OF(IfcFacetedBrep);

Voids : SET [1:?] OF IfcClosedShell;

END_ENTITY;

ENTITY IfcSweptAreaSolid

ABSTRACT SUPERTYPE OF (ONEOF(IfcExtrudedAreaSolid, IfcFixedReferenceSweptAreaSolid,
IfcRevolvedAreaSolid, IfcSurfaceCurveSweptAreaSolid))

SUBTYPE OF(IfcSolidModel);

SweptArea : IfcProfileDef;

Position : OPTIONAL IfcAxis2Placement3D;

WHERE

SWEPTAREATYPE : SweptArea.ProfileType = IfcProfileTypeEnum.Area;

END_ENTITY;

ENTITY IfcExtrudedAreaSolid

```

SUBTYPE OF(IfcSweptAreaSolid);
    ExtrudedDirection : IfcDirection;
    Depth              : IfcPositiveLengthMeasure;
WHERE
    VALIDEXTRUSIONDIRECTION : IfcDotProduct(IfcRepresentationItem() ||
IfcGeometricRepresentationItem() || IfcDirection([0.0,0.0,1.0]), SELF.ExtrudedDirection) <> 0.0;
END_ENTITY;

```

```

ENTITY IfcExtrudedAreaSolidTapered
    SUBTYPE OF(IfcExtrudedAreaSolid);
        EndSweptArea : IfcProfileDef;
WHERE
    CORRECTPROFILEASSIGNMENT : IfcTaperedSweptAreaProfiles(SELF.IfcSweptAreaSolid.SweptArea,
SELF.EndSweptArea);
END_ENTITY;

```

```

ENTITY IfcFixedReferenceSweptAreaSolid
    SUBTYPE OF(IfcSweptAreaSolid);
        Directrix      : IfcCurve;
        StartParam     : OPTIONAL IfcParameterValue;
        EndParam       : OPTIONAL IfcParameterValue;
        FixedReference : IfcDirection;
WHERE
    DIRECTRIXBOUNDED : (EXISTS(StartParam) AND EXISTS(EndParam)) OR
                        (SIZEOF(['IFC4.IFCCONIC', 'IFC4.IFCBOUNDEDCURVE'] * TYPEOF(Directrix)) =
1);
END_ENTITY;

```

```

ENTITY IfcRevolvedAreaSolid
    SUBTYPE OF(IfcSweptAreaSolid);
        Axis : IfcAxis1Placement;

```

```

    Angle      : IfcPlaneAngleMeasure;
DERIVE
    AxisLine : IfcLine := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcCurve() || IfcLine(Axis.Location,
                        IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcVector(Axis.Z,1.0));
WHERE
    AXISSTARTINXY      : Axis.Location.Coordinates[3] = 0.0;
    AXISDIRECTIONINXY : Axis.Z.DirectionRatios[3] = 0.0;
END_ENTITY;

ENTITY IfcRevolvedAreaSolidTapered
    SUBTYPE OF(IfcRevolvedAreaSolid);
    EndSweptArea : IfcProfileDef;
WHERE
    CORRECTPROFILEASSIGNMENT : IfcTaperedSweptAreaProfiles(SELFWIfcSweptAreaSolid.SweptArea,
SELF.EndSweptArea);
END_ENTITY;

ENTITY IfcSurfaceCurveSweptAreaSolid
    SUBTYPE OF(IfcSweptAreaSolid);
    Directrix      : IfcCurve;
    StartParam     : OPTIONAL IfcParameterValue;
    EndParam       : OPTIONAL IfcParameterValue;
    ReferenceSurface : IfcSurface;
WHERE
    DIRECTRIXBOUNDED : (EXISTS(StartParam) AND EXISTS(EndParam)) OR
                        (SIZEOF(['IFC4.IFCCONIC', 'IFC4.IFCBOUNDEDCURVE'] * TYPEOF(Directrix)) =
1);
END_ENTITY;

```

ENTITY IfcSweptDiskSolid

SUBTYPE OF(IfcSolidModel);

Directrix : IfcCurve;

Radius : IfcPositiveLengthMeasure;

InnerRadius : OPTIONAL IfcPositiveLengthMeasure;

StartParam : OPTIONAL IfcParameterValue;

EndParam : OPTIONAL IfcParameterValue;

WHERE

DIRECTRIXDIM : Directrix.Dim = 3;

INNERRADIUSSIZE : (NOT EXISTS(InnerRadius)) OR (Radius > InnerRadius);

DIRECTRIXBOUNDED : (EXISTS(StartParam) AND EXISTS(EndParam)) OR

(SIZEOF(['IFC4.IFCCONIC', 'IFC4.IFCBOUNDEDCURVE'] * TYPEOF(Directrix)) =

1);

END_ENTITY;

ENTITY IfcSweptDiskSolidPolygonal

SUBTYPE OF(IfcSweptDiskSolid);

FilletRadius : OPTIONAL IfcPositiveLengthMeasure;

WHERE

CORRECTRADIUS : NOT(EXISTS(FilletRadius)) OR (FilletRadius >=
SELF.IfcSweptDiskSolid.Radius);

DIRECTRIXISPOLYLINE : 'IFC4.IFCPOLYLINE' IN TYPEOF(SELFW.IfcSweptDiskSolid.Directrix);

END_ENTITY;

ENTITY IfcSpace

SUBTYPE OF(IfcSpatialStructureElement);

PredefinedType : OPTIONAL IfcSpaceTypeEnum;

ElevationWithFlooring : OPTIONAL IfcLengthMeasure;

INVERSE

HasCoverings : SET OF IfcRelCoversSpaces FOR RelatingSpace;

```

BoundedBy          : SET OF IfcRelSpaceBoundary FOR RelatingSpace;
WHERE
CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcSpaceTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcSpaceTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED  : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCSPACETYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSpatialStructureElement

```

```

ABSTRACT SUPERTYPE OF (ONEOF(IfcBuilding, IfcBuildingStorey, IfcSite, IfcSpace))

```

```

SUBTYPE OF(IfcSpatialElement);

```

```

CompositionType : OPTIONAL IfcElementCompositionEnum;

```

```

WHERE

```

```

WR41 : (HINDEX(SELFWIfcObjectDefinition.Decomposes) = 1)

```

```

AND

```

```

('IFC4.IFCRELAGGREGATES' IN TYPEOF(SELFWIfcObjectDefinition.Decomposes[1]))

```

```

AND

```

```

(('IFC4.IFCPROJECT' IN TYPEOF
(SELFWIfcObjectDefinition.Decomposes[1].RelatingObject)) OR

```

```

('IFC4.IFCSPATIALSTRUCTUREELEMENT' IN TYPEOF
(SELFWIfcObjectDefinition.Decomposes[1].RelatingObject))

```

```

);

```

```

END_ENTITY;

```

```

ENTITY IfcBuilding

```

```

SUBTYPE OF(IfcSpatialStructureElement);

```

```

ElevationOfRefHeight : OPTIONAL IfcLengthMeasure;

```

```

ElevationOfTerrain : OPTIONAL IfcLengthMeasure;

```

BuildingAddress : OPTIONAL IfcPostalAddress;
END_ENTITY;

ENTITY IfcBuildingStorey
SUBTYPE OF(IfcSpatialStructureElement);
Elevation : OPTIONAL IfcLengthMeasure;
END_ENTITY;

ENTITY IfcSite
SUBTYPE OF(IfcSpatialStructureElement);
RefLatitude : OPTIONAL IfcCompoundPlaneAngleMeasure;
RefLongitude : OPTIONAL IfcCompoundPlaneAngleMeasure;
RefElevation : OPTIONAL IfcLengthMeasure;
LandTitleNumber : OPTIONAL IfcLabel;
SiteAddress : OPTIONAL IfcPostalAddress;
END_ENTITY;

ENTITY IfcRelCoversSpaces
SUBTYPE OF(IfcRelConnects);
RelatingSpace : IfcSpace;
RelatedCoverings : SET [1:?] OF IfcCovering;
END_ENTITY;

ENTITY IfcCovering
SUBTYPE OF(IfcBuildingElement);
PredefinedType : OPTIONAL IfcCoveringTypeEnum;
INVERSE
CoversSpaces : SET [0:1] OF IfcRelCoversSpaces FOR RelatedCoverings;
CoversElements : SET [0:1] OF IfcRelCoversBldgElements FOR RelatedCoverings;
WHERE


```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcCoveringTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcCoveringTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                      ('IFC4.IFCCOVERINGTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcBuildingElement

```

```

ABSTRACT SUPERTYPE OF (ONEOF(IfcBeam, IfcBuildingElementProxy, IfcChimney, IfcColumn,
IfcCovering, IfcCurtainWall, IfcDoor, IfcFooting, IfcMember, IfcPile, IfcPlate, IfcRailing,
IfcRamp, IfcRampFlight, IfcRoof, IfcShadingDevice, IfcSlab, IfcStair, IfcStairFlight, IfcWall,
IfcWindow))

```

```

SUBTYPE OF(IfcElement);

```

```

WHERE

```

```

MAXONEMATERIALASSOCIATION : SIZEOF (QUERY(temp < * SELFWIfcObjectDefinition.HasAssociations
|
                        'IFC4.IFCREASSOCIATESMATERIAL' IN TYPEOF(temp)
                        )) <= 1;

```

```

END_ENTITY;

```

```

ENTITY IfcBeam

```

```

SUBTYPE OF(IfcBuildingElement);

```

```

PredefinedType : OPTIONAL IfcBeamTypeEnum;

```

```

WHERE

```

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcBeamTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcBeamTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                      ('IFC4.IFCBEAMTYPE' IN

```

```

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

ENTITY IfcBeamStandardCase
  SUBTYPE OF(IfcBeam);
  WHERE
    HASMATERIALPROFILESETUSAGE : SIZEOF (QUERY(temp < * USEDIN(SELF,
'IFC4.IFCRELASSOCIATES.RELATEDOBJECTS') |
                                ('IFC4.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp)) AND
                                ('IFC4.IFCMATERIALPROFILESETUSAGE' IN
TYPEOF(temp.RelatingMaterial))
                                )) = 1;
END_ENTITY;

ENTITY IfcBuildingElementProxy
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcBuildingElementProxyTypeEnum;
  WHERE
    HASOBJECTNAME : EXISTS(SELFWIfcRoot.Name);
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcBuildingElementProxyTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcBuildingElementProxyTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          ('IFC4.IFCBUILDINGELEMENTPROXYTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

ENTITY IfcChimney
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcChimneyTypeEnum;

```

WHERE

```
CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcChimneyTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcChimneyTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCCHIMNEYTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

ENTITY IfcColumn

SUBTYPE OF(IfcBuildingElement);

PredefinedType : OPTIONAL IfcColumnTypeEnum;

WHERE

```
CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcColumnTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcColumnTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCCOLUMNTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

ENTITY IfcColumnStandardCase

SUBTYPE OF(IfcColumn);

WHERE

```
HASMATERIALPROFILESETUSAGE : SIZEOF (QUERY(temp < * USEDIN(SELF,
'IFC4.IFCREASSOCIATES.RELATEDOBJECTS') |
                        ('IFC4.IFCREASSOCIATESMATERIAL' IN TYPEOF(temp)) AND
                        ('IFC4.IFCMATERIALPROFILESETUSAGE' IN
TYPEOF(temp.RelatingMaterial))
                        )) = 1;
```

END_ENTITY;

ENTITY IfcCurtainWall

SUBTYPE OF(IfcBuildingElement);

PredefinedType : OPTIONAL IfcCurtainWallTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcCurtainWallTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCurtainWallTypeEnum.USERDEFINED) AND EXISTS

(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCCURTAINWALLTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcDoor

SUBTYPE OF(IfcBuildingElement);

OverallHeight : OPTIONAL IfcPositiveLengthMeasure;

OverallWidth : OPTIONAL IfcPositiveLengthMeasure;

PredefinedType : OPTIONAL IfcDoorTypeEnum;

OperationType : OPTIONAL IfcDoorTypeOperationEnum;

UserDefinedOperationType : OPTIONAL IfcLabel;

WHERE

CORRECTSTYLEASSIGNED : (SIZEOF(IsTypedBy) = 0)

OR

('IFC4.IFCDOORTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcDoorStandardCase

SUBTYPE OF(IfcDoor);

END_ENTITY;

ENTITY IfcFooting

SUBTYPE OF(IfcBuildingElement);

PredefinedType : OPTIONAL IfcFootingTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT EXISTS(PredefinedType) OR

(PredefinedType <> IfcFootingTypeEnum.USERDEFINED) OR

((PredefinedType = IfcFootingTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCFOOTINGTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcMember

SUBTYPE OF(IfcBuildingElement);

PredefinedType : OPTIONAL IfcMemberTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcMemberTypeEnum.USERDEFINED) OR

((PredefinedType = IfcMemberTypeEnum.USERDEFINED) AND EXISTS

(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCMEMBERTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcMemberStandardCase

SUBTYPE OF(IfcMember);

WHERE

HASMATERIALPROFILESETUSAGE : SIZEOF (QUERY(temp < * USEDIN(SELF,

```

' IFC4. IFCRECLASSOCIATES.RELATEDOBJECTS' ) |
                                ( ' IFC4. IFCRECLASSOCIATESMATERIAL' IN TYPEOF(temp)) AND
                                ( ' IFC4. IFCMATERIALPROFILESETUSAGE' IN
TYPEOF(temp.RelatingMaterial))
                                )) = 1;

END_ENTITY;

```

```

ENTITY IfcPile
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcPileTypeEnum;
  ConstructionType : OPTIONAL IfcPileConstructionEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT EXISTS(PredefinedType) OR
                            (PredefinedType <> IfcPileTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcPileTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                            ( ' IFC4. IFCPILETYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

```

```

ENTITY IfcPlate
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcPlateTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                            (PredefinedType <> IfcPlateTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcPlateTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                            ( ' IFC4. IFCPLATETYPE' IN

```

```

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

ENTITY IfcPlateStandardCase
  SUBTYPE OF(IfcPlate);
  WHERE
    HASMATERIALLAYERSETUSAGE : SIZEOF (QUERY(temp <* USEDIN(SELF,
'IFC4.IFCREASSOCIATES.RELATEDOBJECTS') |
                                ('IFC4.IFCREASSOCIATESMATERIAL' IN TYPEOF(temp)) AND
                                ('IFC4.IFCMATERIALLAYERSETUSAGE' IN
TYPEOF(temp.RelatingMaterial))
                                )) = 1;
END_ENTITY;

ENTITY IfcRailing
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcRailingTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcRailingTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcRailingTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          ('IFC4.IFCRAILINGTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

ENTITY IfcRamp
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcRampTypeEnum;
  WHERE

```

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcRampTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcRampTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCRAMPTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcRampFlight

SUBTYPE OF(IfcBuildingElement);

PredefinedType : OPTIONAL IfcRampFlightTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcRampFlightTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcRampFlightTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCRAMPFLIGHTTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcRoof

SUBTYPE OF(IfcBuildingElement);

PredefinedType : OPTIONAL IfcRoofTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcRoofTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcRoofTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```



```

                (' IFC4.IFCROOFTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcShadingDevice
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcShadingDeviceTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcShadingDeviceTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcShadingDeviceTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          (' IFC4.IFCSHADINGDEVICETYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSlab
  SUPERTYPE OF (ONEOF(IfcSlabElementedCase, IfcSlabStandardCase))
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcSlabTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcSlabTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcSlabTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          (' IFC4.IFCSLABTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSlabElementedCase
  SUBTYPE OF(IfcSlab);
  WHERE
    HASDECOMPOSITION : HI INDEX(SELFWIfcObjectDefinition.IsDecomposedBy) > 0;
END_ENTITY;

```

```

ENTITY IfcSlabStandardCase
  SUBTYPE OF(IfcSlab);
  WHERE
    HASMATERIALLAYERSETUSAGE : SIZEOF (QUERY(temp < * USEDIN(SELF,
'IFC4.IFCREASSOCIATES.RELATEDOBJECTS') |
                                ('IFC4.IFCREASSOCIATESMATERIAL' IN TYPEOF(temp)) AND
                                ('IFC4.IFCMATERIALLAYERSETUSAGE' IN
TYPEOF(temp.RelatingMaterial))
                                )) = 1;
END_ENTITY;

```

```

ENTITY IfcStair
  SUBTYPE OF(IfcBuildingElement);
  PredefinedType : OPTIONAL IfcStairTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcStairTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcStairTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          ('IFC4.IFCSTAIRTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcStairFlight

```

```

SUBTYPE OF(IfcBuildingElement);
    NumberOfRisers : OPTIONAL IfcInteger;
    NumberOfTreads : OPTIONAL IfcInteger;
    RiserHeight    : OPTIONAL IfcPositiveLengthMeasure;
    TreadLength    : OPTIONAL IfcPositiveLengthMeasure;
    PredefinedType : OPTIONAL IfcStairFlightTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcStairFlightTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcStairFlightTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED    : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCSTAIRFLIGHTTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcWall
    SUPERTYPE OF (ONEOF(IfcWallElementedCase, IfcWallStandardCase))
    SUBTYPE OF(IfcBuildingElement);
    PredefinedType : OPTIONAL IfcWallTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcWallTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcWallTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED    : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCWALLTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcWallElementedCase

```

```

SUBTYPE OF(IfcWall);
WHERE
    HASDECOMPOSITION : HI INDEX(SELFWIfcObjectDefinition.IsDecomposedBy) > 0;
END_ENTITY;

```

```

ENTITY IfcWallStandardCase
    SUBTYPE OF(IfcWall);
    WHERE
        HASMATERIALLAYERSETUSAGE : SIZEOF (QUERY(temp <* USEDIN(SELF,
'IFC4.IFCREASSOCIATES.RELATEDOBJECTS') |
                                ('IFC4.IFCREASSOCIATESMATERIAL' IN TYPEOF(temp)) AND
                                ('IFC4.IFCMATERIALLAYERSETUSAGE' IN
TYPEOF(temp.RelatingMaterial))
                                )) = 1;
    END_ENTITY;

```

```

ENTITY IfcWindow
    SUBTYPE OF(IfcBuildingElement);
    OverallHeight : OPTIONAL IfcPositiveLengthMeasure;
    OverallWidth : OPTIONAL IfcPositiveLengthMeasure;
    PredefinedType : OPTIONAL IfcWindowTypeEnum;
    PartitioningType : OPTIONAL IfcWindowTypePartitioningEnum;
    UserDefinedPartitioningType : OPTIONAL IfcLabel;
    WHERE
        CORRECTSTYLEASSIGNED : (SIZEOF(IsTypedBy) = 0)
        OR ('IFC4.IFCWINDOWTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
    END_ENTITY;

```

```

ENTITY IfcWindowStandardCase
    SUBTYPE OF(IfcWindow);

```

END_ENTITY;

ENTITY IfcRelCoversBldgElements

SUBTYPE OF(IfcRelConnects);

RelatingBuildingElement : IfcElement;

RelatedCoverings : SET [1:?] OF IfcCovering;

END_ENTITY;

ENTITY IfcSpatialZone

SUBTYPE OF(IfcSpatialElement);

PredefinedType : OPTIONAL IfcSpatialZoneTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcSpatialZoneTypeEnum.USERDEFINED) OR

((PredefinedType = IfcSpatialZoneTypeEnum.USERDEFINED) AND EXISTS

(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCSPATIALZONETYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcCivilSpatialStructureElement_K

ABSTRACT SUPERTYPE OF (ONEOF(IfcRoad_K, IfcBridge_K, IfcTunnel_K))

SUBTYPE OF(IfcSpatialElement);

END_ENTITY;

ENTITY IfcRoad_K

SUBTYPE OF(IfcCivilSpatialStructureElement_K);

PredefinedType : OPTIONAL IfcRoadTypeEnum_K;

END_ENTITY;

```
ENTITY IfcBridge_K
  SUBTYPE OF(IfcCivilSpatialStructureElement_K);
  PredefinedType : OPTIONAL IfcBridgeTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcTunnel_K
  SUBTYPE OF(IfcCivilSpatialStructureElement_K);
  PredefinedType : OPTIONAL IfcTunnelTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcCivilSpatialBoundary_K
  ABSTRACT SUPERTYPE OF (ONEOF(IfcLinearRefSpace_K, IfcCurvilinearNodeSpace_K,
IfcVerticalSubspace_K))
  SUBTYPE OF(IfcSpatialElement);
END_ENTITY;
```

```
ENTITY IfcLinearRefSpace_K
  SUBTYPE OF(IfcCivilSpatialBoundary_K);
  PredefinedType : OPTIONAL IfcLinearRefSpaceTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcCurvilinearNodeSpace_K
  SUBTYPE OF(IfcCivilSpatialBoundary_K);
  PredefinedType : OPTIONAL IfcCurvilinearNodeSpaceTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcVerticalSubspace_K
  SUBTYPE OF(IfcCivilSpatialBoundary_K);
  PredefinedType : OPTIONAL IfcVerticalSubspaceTypeEnum_K;
```

END_ENTITY;

ENTITY IfcRelReferencedInSpatialStructure

SUBTYPE OF(IfcRelConnects);

RelatedElements : SET [1:?] OF IfcProduct;

RelatingStructure : IfcSpatialElement;

WHERE

WR31 : SIZEOF(QUERY(temp <* RelatedElements | 'IFC4.IFCSPATIALSTRUCTUREELEMENT' IN
TYPEOF(temp))) = 0;

END_ENTITY;

ENTITY IfcGrid

SUBTYPE OF(IfcProduct);

UAxes : LIST [1:?] OF IfcGridAxis;

VAxes : LIST [1:?] OF IfcGridAxis;

WAxes : OPTIONAL LIST [1:?] OF IfcGridAxis;

PredefinedType : OPTIONAL IfcGridTypeEnum;

INVERSE

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

WHERE

HASPLACEMENT : EXISTS(SELFWIfcProduct.ObjectPlacement);

END_ENTITY;

ENTITY IfcGridAxis;

AxisTag : OPTIONAL IfcLabel;

AxisCurve : IfcCurve;

SameSense : IfcBoolean;

INVERSE

PartOfW : SET [0:1] OF IfcGrid FOR WAxes;

PartOfV : SET [0:1] OF IfcGrid FOR VAxes;

```

PartOfU          : SET [0:1] OF IfcGrid FOR UAxes;
HasIntersections : SET OF IfcVirtualGridIntersection FOR IntersectingAxes;
WHERE
  WR1 : AxisCurve.Dim = 2;
  WR2 : (SIZEOF(PartOfU) = 1) XOR (SIZEOF(PartOfV) = 1) XOR (SIZEOF(PartOfW) = 1);
END_ENTITY;

```

```

ENTITY IfcVirtualGridIntersection;
  IntersectingAxes : LIST [2:2] OF IfcGridAxis;
  OffsetDistances  : LIST [2:3] OF IfcLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcPort
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcProduct);
  INVERSE
    ContainedIn   : SET [0:1] OF IfcRelConnectsPortToElement FOR RelatingPort;
    ConnectedFrom : SET [0:1] OF IfcRelConnectsPorts FOR RelatedPort;
    ConnectedTo   : SET [0:1] OF IfcRelConnectsPorts FOR RelatingPort;
END_ENTITY;

```

```

ENTITY IfcDistributionPort
  SUBTYPE OF(IfcPort);
  FlowDirection : OPTIONAL IfcFlowDirectionEnum;
  PredefinedType : OPTIONAL IfcDistributionPortTypeEnum;
  SystemType     : OPTIONAL IfcDistributionSystemEnum;
END_ENTITY;

```

```

ENTITY IfcRelConnectsPortToElement
  SUBTYPE OF(IfcRelConnects);

```


RelatingPort : IfcPort;
RelatedElement : IfcDistributionElement;

END_ENTITY;

ENTITY IfcDistributionElement

SUPERTYPE OF (ONEOF(IfcDistributionControlElement, IfcDistributionFlowElement))

SUBTYPE OF(IfcElement);

INVERSE

HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;

END_ENTITY;

ENTITY IfcDistributionControlElement

SUPERTYPE OF (ONEOF(IfcActuator, IfcAlarm, IfcController, IfcFlowInstrument,
IfcProtectiveDeviceTrippingUnit, IfcSensor, IfcUnitaryControlElement))

SUBTYPE OF(IfcDistributionElement);

INVERSE

AssignedToFlowElement : SET [0:1] OF IfcRelFlowControlElements FOR RelatedControlElements;

END_ENTITY;

ENTITY IfcActuator

SUBTYPE OF(IfcDistributionControlElement);

PredefinedType : OPTIONAL IfcActuatorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcActuatorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcActuatorTypeEnum.USERDEFINED) AND EXISTS

(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCACTUATORTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

```

ENTITY IfcAlarm
  SUBTYPE OF(IfcDistributionControlElement);
  PredefinedType : OPTIONAL IfcAlarmTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcAlarmTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcAlarmTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCALARMTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcController
  SUBTYPE OF(IfcDistributionControlElement);
  PredefinedType : OPTIONAL IfcControllerTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcControllerTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcControllerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCCONTROLLERTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcFlowInstrument
  SUBTYPE OF(IfcDistributionControlElement);
  PredefinedType : OPTIONAL IfcFlowInstrumentTypeEnum;
  WHERE

```

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcFlowInstrumentTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcFlowInstrumentTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCFLOWINSTRUMENTTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcProtectiveDeviceTrippingUnit

SUBTYPE OF(IfcDistributionControlElement);

PredefinedType : OPTIONAL IfcProtectiveDeviceTrippingUnitTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcProtectiveDeviceTrippingUnitTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcProtectiveDeviceTrippingUnitTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCPROTECTIVEDEVICETRIPPINGUNITTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcSensor

SUBTYPE OF(IfcDistributionControlElement);

PredefinedType : OPTIONAL IfcSensorTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcSensorTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcSensorTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```

```

                ('IFC4.IFCSENSORTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcUnitaryControlElement
  SUBTYPE OF(IfcDistributionControlElement);
  PredefinedType : OPTIONAL IfcUnitaryControlElementTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcUnitaryControlElementTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcUnitaryControlElementTypeEnum.USERDEFINED)
AND EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCUNITARYCONTROLELEMENTTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcReIFlowControlElements
  SUBTYPE OF(IfcReIConnects);
  RelatedControlElements : SET [1:?] OF IfcDistributionControlElement;
  RelatingFlowElement : IfcDistributionFlowElement;
END_ENTITY;

```

```

ENTITY IfcDistributionFlowElement
  SUPERTYPE OF (ONEOF(IfcDistributionChamberElement, IfcEnergyConversionDevice,
IfcFlowController, IfcFlowFitting, IfcFlowMovingDevice, IfcFlowSegment, IfcFlowStorageDevice,
IfcFlowTerminal, IfcFlowTreatmentDevice))
  SUBTYPE OF(IfcDistributionElement);
  INVERSE
    HasControlElements : SET [0:1] OF IfcReIFlowControlElements FOR RelatingFlowElement;
END_ENTITY;

```

```

ENTITY IfcDistributionChamberElement
    SUBTYPE OF(IfcDistributionFlowElement);
    PredefinedType : OPTIONAL IfcDistributionChamberElementTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
            (PredefinedType <> IfcDistributionChamberElementTypeEnum.USERDEFINED) OR
            ((PredefinedType = IfcDistributionChamberElementTypeEnum.USERDEFINED) AND EXISTS (SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
            ('IFC4.IFCDISTRIBUTIONCHAMBERELEMENTTYPE' IN
            TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
    END_ENTITY;

```

```

ENTITY IfcEnergyConversionDevice
    SUPERTYPE OF (ONEOF(IfcAirToAirHeatRecovery, IfcBoiler, IfcBurner, IfcChiller, IfcCoil,
    IfcCondenser, IfcCooledBeam, IfcCoolingTower, IfcElectricGenerator, IfcElectricMotor, IfcEngine,
    IfcEvaporativeCooler, IfcEvaporator, IfcHeatExchanger, IfcHumidifier, IfcMotorConnection,
    IfcSolarDevice, IfcTransformer, IfcTubeBundle, IfcUnitaryEquipment))
    SUBTYPE OF(IfcDistributionFlowElement);
    END_ENTITY;

```

```

ENTITY IfcAirToAirHeatRecovery
    SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcAirToAirHeatRecoveryTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
            (PredefinedType <> IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) OR
            ((PredefinedType = IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) AND
            EXISTS (SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```

```

                ('IFC4.IFCAIRTOAIRHEATRECOVERYTYPE'
                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcBoiler
  SUBTYPE OF(IfcEnergyConversionDevice);
  PredefinedType : OPTIONAL IfcBoilerTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcBoilerTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcBoilerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          ('IFC4.IFCBOILERTYPE'
                          IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcBurner
  SUBTYPE OF(IfcEnergyConversionDevice);
  PredefinedType : OPTIONAL IfcBurnerTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcBurnerTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcBurnerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          ('IFC4.IFCBURNERTYPE'
                          IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcChiller

```

```

SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcChillerTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcChillerTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcChillerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCCHILLERTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcCoil
SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcCoilTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcCoilTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcCoilTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCCOILTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcCondenser
SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcCondenserTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcCondenserTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcCondenserTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCCONDENSERTYPE'
                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcCooledBeam

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcCooledBeamTypeEnum;

WHERE

```

        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                (PredefinedType <> IfcCooledBeamTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcCooledBeamTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCCOOLEDBEAMTYPE'
                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcCoolingTower

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcCoolingTowerTypeEnum;

WHERE

```

        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                (PredefinedType <> IfcCoolingTowerTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcCoolingTowerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCCOOLINGTOWERTYPE'
                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```


ENTITY IfcElectricGenerator

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcElectricGeneratorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcElectricGeneratorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcElectricGeneratorTypeEnum.USERDEFINED) AND

EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCELECTRICGENERATORTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcElectricMotor

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcElectricMotorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcElectricMotorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcElectricMotorTypeEnum.USERDEFINED) AND EXISTS

(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCELECTRICMOTORTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcEngine

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcEngineTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcEngineTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcEngineTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCENGINETYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcEvaporativeCooler

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcEvaporativeCoolerTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcEvaporativeCoolerTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcEvaporativeCoolerTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCEVAPORATIVECOOLERTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcEvaporator

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcEvaporatorTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcEvaporatorTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcEvaporatorTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```

```

                (' IFC4.IFCEVAPORATORTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcHeatExchanger
  SUBTYPE OF(IfcEnergyConversionDevice);
  PredefinedType : OPTIONAL IfcHeatExchangerTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcHeatExchangerTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcHeatExchangerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          (' IFC4.IFCHEATEXCHANGERTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcHumidifier
  SUBTYPE OF(IfcEnergyConversionDevice);
  PredefinedType : OPTIONAL IfcHumidifierTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcHumidifierTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcHumidifierTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          (' IFC4.IFCHUMIDIFIERTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcMotorConnection

```

```

SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcMotorConnectionTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcMotorConnectionTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcMotorConnectionTypeEnum.USERDEFINED) AND
EXISTS (SELFIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCMOTORCONNECTIONTYPE'
                                IN
TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSolarDevice
SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcSolarDeviceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcSolarDeviceTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcSolarDeviceTypeEnum.USERDEFINED) AND EXISTS
(SELFIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCSOLARDEVICETYPE'
                                IN
TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcTransformer
SUBTYPE OF(IfcEnergyConversionDevice);
    PredefinedType : OPTIONAL IfcTransformerTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcTransformerTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcTransformerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCTRANFORMERTYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcTubeBundle

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcTubeBundleTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                (PredefinedType <> IfcTubeBundleTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcTubeBundleTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCTUBE BUNDLETYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcUnitaryEquipment

SUBTYPE OF(IfcEnergyConversionDevice);

PredefinedType : OPTIONAL IfcUnitaryEquipmentTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                (PredefinedType <> IfcUnitaryEquipmentTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcUnitaryEquipmentTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCUNITARYEQUIPMENTTYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcFlowController

SUPERTYPE OF (ONEOF(IfcAirTerminalBox, IfcDamper, IfcElectricDistributionBoard, IfcElectricTimeControl, IfcFlowMeter, IfcProtectiveDevice, IfcSwitchingDevice, IfcValve))

SUBTYPE OF(IfcDistributionFlowElement);

END_ENTITY;

ENTITY IfcAirTerminalBox

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcAirTerminalBoxTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcAirTerminalBoxTypeEnum.USERDEFINED) OR

((PredefinedType = IfcAirTerminalBoxTypeEnum.USERDEFINED) AND EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

(' IFC4.IFCAIRTERMINALBOXTYPE' IN TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcDamper

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcDamperTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcDamperTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDamperTypeEnum.USERDEFINED) AND EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

(' IFC4.IFCDAMPERTYPE' IN TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcElectricDistributionBoard

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcElectricDistributionBoardTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcElectricDistributionBoardTypeEnum.USERDEFINED)

OR

((PredefinedType = IfcElectricDistributionBoardTypeEnum.USERDEFINED)

AND EXISTS (SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCELECTRICDISTRIBUTIONBOARDTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcElectricTimeControl

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcElectricTimeControlTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcElectricTimeControlTypeEnum.USERDEFINED) OR

((PredefinedType = IfcElectricTimeControlTypeEnum.USERDEFINED) AND

EXISTS (SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCELECTRICTIMECONTROLTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcFlowMeter

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcFlowMeterTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcFlowMeterTypeEnum.USERDEFINED) OR

((PredefinedType = IfcFlowMeterTypeEnum.USERDEFINED) AND EXISTS
(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCFLOWMETERTYPE' IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcProtectiveDevice

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcProtectiveDeviceTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcProtectiveDeviceTypeEnum.USERDEFINED) OR

((PredefinedType = IfcProtectiveDeviceTypeEnum.USERDEFINED) AND
EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCPROTECTIVEDEVICETYPE' IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcSwitchingDevice

SUBTYPE OF(IfcFlowController);

PredefinedType : OPTIONAL IfcSwitchingDeviceTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcSwitchingDeviceTypeEnum.USERDEFINED) OR

((PredefinedType = IfcSwitchingDeviceTypeEnum.USERDEFINED) AND


```

EXISTS (SELFIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          ('IFC4.IFCSWITCHINGDEVICETYPE'
                           IN
TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcValve

```

```

    SUBTYPE OF (IfcFlowController);

```

```

    PredefinedType : OPTIONAL IfcValveTypeEnum;

```

```

    WHERE

```

```

        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

```

```

            (PredefinedType <> IfcValveTypeEnum.USERDEFINED) OR

```

```

            ((PredefinedType = IfcValveTypeEnum.USERDEFINED) AND EXISTS
(SELFIfcObject.ObjectType));

```

```

    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```

```

            ('IFC4.IFCVALVETYPE'

```

```

            IN
TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

```

```

END_ENTITY;

```

```

ENTITY IfcFlowFitting

```

```

    SUPERTYPE OF (ONEOF(IfcCableCarrierFitting, IfcCableFitting, IfcDuctFitting, IfcJunctionBox,
IfcPipeFitting, IfcGutterFitting_K))

```

```

    SUBTYPE OF (IfcDistributionFlowElement);

```

```

END_ENTITY;

```

```

ENTITY IfcCableCarrierFitting

```

```

    SUBTYPE OF (IfcFlowFitting);

```

```

    PredefinedType : OPTIONAL IfcCableCarrierFittingTypeEnum;

```

```

    WHERE

```

```

        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

```

```

            (PredefinedType <> IfcCableCarrierFittingTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcCableCarrierFittingTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCCABLECARRIERFITTINGTYPE'
                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcCableFitting

SUBTYPE OF(IfcFlowFitting);

PredefinedType : OPTIONAL IfcCableFittingTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcCableFittingTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCableFittingTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCCABLEFITTINGTYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcDuctFitting

SUBTYPE OF(IfcFlowFitting);

PredefinedType : OPTIONAL IfcDuctFittingTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcDuctFittingTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDuctFittingTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCDUCTFITTINGTYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

```

ENTITY IfcJunctionBox
  SUBTYPE OF(IfcFlowFitting);
  PredefinedType : OPTIONAL IfcJunctionBoxTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcJunctionBoxTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcJunctionBoxTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCJUNCTIONBOXTYPE'
                                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcPipeFitting
  SUBTYPE OF(IfcFlowFitting);
  PredefinedType : OPTIONAL IfcPipeFittingTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcPipeFittingTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcPipeFittingTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCPIPEFITTINGTYPE'
                                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcGutterFitting_K
  SUBTYPE OF(IfcFlowFitting);
  PredefinedType : OPTIONAL IfcGutterFittingTypeEnum_K;
  END_ENTITY;

```

ENTITY IfcFlowMovingDevice

SUPERTYPE OF (ONEOF(IfcCompressor, IfcFan, IfcPump))

SUBTYPE OF(IfcDistributionFlowElement);

END_ENTITY;

ENTITY IfcCompressor

SUBTYPE OF(IfcFlowMovingDevice);

PredefinedType : OPTIONAL IfcCompressorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcCompressorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCompressorTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCCOMPRESSORTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcFan

SUBTYPE OF(IfcFlowMovingDevice);

PredefinedType : OPTIONAL IfcFanTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcFanTypeEnum.USERDEFINED) OR

((PredefinedType = IfcFanTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCFANTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

```

ENTITY IfcPump
  SUBTYPE OF(IfcFlowMovingDevice);
  PredefinedType : OPTIONAL IfcPumpTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcPumpTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcPumpTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCPUMPTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcFlowSegment
  SUPERTYPE OF (ONEOF(IfcCableCarrierSegment, IfcCableSegment, IfcDuctSegment, IfcPipeSegment,
IfcGutterSegment_K))
  SUBTYPE OF(IfcDistributionFlowElement);
  END_ENTITY;

```

```

ENTITY IfcCableCarrierSegment
  SUBTYPE OF(IfcFlowSegment);
  PredefinedType : OPTIONAL IfcCableCarrierSegmentTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcCableCarrierSegmentTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcCableCarrierSegmentTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCCABLECARRIERSEGMENTTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

```

END_ENTITY;

ENTITY IfcCableSegment

SUBTYPE OF(IfcFlowSegment);

PredefinedType : OPTIONAL IfcCableSegmentTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcCableSegmentTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCableSegmentTypeEnum.USERDEFINED) AND EXISTS

(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCCABLESEGMENTTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcDuctSegment

SUBTYPE OF(IfcFlowSegment);

PredefinedType : OPTIONAL IfcDuctSegmentTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcDuctSegmentTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDuctSegmentTypeEnum.USERDEFINED) AND EXISTS

(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCDUCTSEGMENTTYPE'

IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcPipeSegment

SUBTYPE OF(IfcFlowSegment);

PredefinedType : OPTIONAL IfcPipeSegmentTypeEnum;

```

WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                            (PredefinedType <> IfcPipeSegmentTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcPipeSegmentTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                            ('IFC4.IFCPIPESEGMENTTYPE'
                            IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

ENTITY IfcGutterSegment_K
    SUBTYPE OF(IfcFlowSegment);
    PredefinedType : OPTIONAL IfcGutterSegmentTypeEnum_K;
END_ENTITY;

ENTITY IfcFlowStorageDevice
    SUPERTYPE OF (ONEOF(IfcElectricFlowStorageDevice, IfcTank))
    SUBTYPE OF(IfcDistributionFlowElement);
END_ENTITY;

ENTITY IfcElectricFlowStorageDevice
    SUBTYPE OF(IfcFlowStorageDevice);
    PredefinedType : OPTIONAL IfcElectricFlowStorageDeviceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                            (PredefinedType <> IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED)
OR
                            ((PredefinedType = IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED)
AND EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                            ('IFC4.IFCELECTRICFLOWSTORAGEDEVICETYPE'
                            IN

```

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcTank

SUBTYPE OF(IfcFlowStorageDevice);

PredefinedType : OPTIONAL IfcTankTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcTankTypeEnum.USERDEFINED) OR

((PredefinedType = IfcTankTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCTANKTYPE' IN

TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcFlowTerminal

SUPERTYPE OF (ONEOF(IfcAirTerminal, IfcAudioVisualAppliance, IfcCommunicationsAppliance,
IfcElectricAppliance, IfcFireSuppressionTerminal, IfcLamp, IfcLightFixture, IfcMedicalDevice,
IfcOutlet, IfcSanitaryTerminal, IfcSpaceHeater, IfcStackTerminal, IfcWasteTerminal))

SUBTYPE OF(IfcDistributionFlowElement);

END_ENTITY;

ENTITY IfcAirTerminal

SUBTYPE OF(IfcFlowTerminal);

PredefinedType : OPTIONAL IfcAirTerminalTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcAirTerminalTypeEnum.USERDEFINED) OR

((PredefinedType = IfcAirTerminalTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));


```

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCAIRTERMINALTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcAudioVisualAppliance
SUBTYPE OF(IfcFlowTerminal);
    PredefinedType : OPTIONAL IfcAudioVisualApplianceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                            (PredefinedType <> IfcAudioVisualApplianceTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcAudioVisualApplianceTypeEnum.USERDEFINED) AND
EXISTIS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                            ('IFC4.IFCAUDIOVISUALAPPLIANCETYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcCommunicationsAppliance
SUBTYPE OF(IfcFlowTerminal);
    PredefinedType : OPTIONAL IfcCommunicationsApplianceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                            (PredefinedType <> IfcCommunicationsApplianceTypeEnum.USERDEFINED)
OR
                            ((PredefinedType = IfcCommunicationsApplianceTypeEnum.USERDEFINED)
AND EXISTIS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                            ('IFC4.IFCCOMMUNICATIONSAPPLIANCETYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcElectricAppliance
  SUBTYPE OF(IfcFlowTerminal);
  PredefinedType : OPTIONAL IfcElectricApplianceTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
      (PredefinedType <> IfcElectricApplianceTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcElectricApplianceTypeEnum.USERDEFINED) AND
    EXISTS (SELFIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
      ('IFC4.IFCELECTRICAPPLIANCETYPE' IN
    TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcFireSuppressionTerminal
  SUBTYPE OF(IfcFlowTerminal);
  PredefinedType : OPTIONAL IfcFireSuppressionTerminalTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
      (PredefinedType <> IfcFireSuppressionTerminalTypeEnum.USERDEFINED)
    OR
      ((PredefinedType = IfcFireSuppressionTerminalTypeEnum.USERDEFINED)
    AND EXISTS (SELFIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
      ('IFC4.IFCFIRESUPPRESSIONTERMINALTYPE' IN
    TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));
  END_ENTITY;

```

```

ENTITY IfcLamp
  SUBTYPE OF(IfcFlowTerminal);
  PredefinedType : OPTIONAL IfcLampTypeEnum;
  WHERE

```

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcLampTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcLampTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCLAMPTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcLightFixture

SUBTYPE OF(IfcFlowTerminal);

PredefinedType : OPTIONAL IfcLightFixtureTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcLightFixtureTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcLightFixtureTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCLIGHTFIXTURETYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcMedicalDevice

SUBTYPE OF(IfcFlowTerminal);

PredefinedType : OPTIONAL IfcMedicalDeviceTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcMedicalDeviceTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcMedicalDeviceTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR

```

```

                (' IFC4.IFCMEDICALDEVICETYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcOutlet
  SUBTYPE OF(IfcFlowTerminal);
  PredefinedType : OPTIONAL IfcOutletTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcOutletTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcOutletTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          (' IFC4.IFCOUTLETTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSanitaryTerminal
  SUBTYPE OF(IfcFlowTerminal);
  PredefinedType : OPTIONAL IfcSanitaryTerminalTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                          (PredefinedType <> IfcSanitaryTerminalTypeEnum.USERDEFINED) OR
                          ((PredefinedType = IfcSanitaryTerminalTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                          (' IFC4.IFCSANITARYTERMINALTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSpaceHeater

```

```

SUBTYPE OF(IfcFlowTerminal);
    PredefinedType : OPTIONAL IfcSpaceHeaterTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcSpaceHeaterTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcSpaceHeaterTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCSPACEHEATERTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcStackTerminal
SUBTYPE OF(IfcFlowTerminal);
    PredefinedType : OPTIONAL IfcStackTerminalTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcStackTerminalTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcStackTerminalTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCSTACKTERMINALTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcWasteTerminal
SUBTYPE OF(IfcFlowTerminal);
    PredefinedType : OPTIONAL IfcWasteTerminalTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcWasteTerminalTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcWasteTerminalTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCWASTETERMINALTYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcFlowTreatmentDevice
    SUPERTYPE OF (ONEOF(IfcDuctSilencer, IfcFilter, IfcInterceptor))
    SUBTYPE OF(IfcDistributionFlowElement);
END_ENTITY;

```

```

ENTITY IfcDuctSilencer
    SUBTYPE OF(IfcFlowTreatmentDevice);
    PredefinedType : OPTIONAL IfcDuctSilencerTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                (PredefinedType <> IfcDuctSilencerTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcDuctSilencerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCDUCTSILENCERTYPE'
IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcFilter
    SUBTYPE OF(IfcFlowTreatmentDevice);
    PredefinedType : OPTIONAL IfcFilterTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                (PredefinedType <> IfcFilterTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcFilterTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCFILTERTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
        END_ENTITY;

```

ENTITY IfcInterceptor

SUBTYPE OF(IfcFlowTreatmentDevice);

PredefinedType : OPTIONAL IfcInterceptorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcInterceptorTypeEnum.USERDEFINED) OR

```

                ((PredefinedType = IfcInterceptorTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));

```

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```

                ('IFC4.IFCINTERCEPTORTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));

```

END_ENTITY;

ENTITY IfcRelConnectsPorts

SUBTYPE OF(IfcRelConnects);

RelatingPort : IfcPort;

RelatedPort : IfcPort;

RealizingElement : OPTIONAL IfcElement;

WHERE

NOSELFREFERENCE : RelatingPort :<>: RelatedPort;

END_ENTITY;

ENTITY IfcProxy

SUBTYPE OF(IfcProduct);

```

ProxyType : IfcObjectTypeEnum;
Tag       : OPTIONAL IfcLabel;
WHERE
    WR1 : EXISTS(SELFWIfcRoot.Name);
END_ENTITY;

```

```

ENTITY IfcStructuralActivity

```

```

    ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralAction, IfcStructuralReaction))
    SUBTYPE OF(IfcProduct);

```

```

    AppliedLoad          : IfcStructuralLoad;
    GlobalOrLocal        : IfcGlobalOrLocalEnum;

```

```

    INVERSE

```

```

        AssignedToStructuralItem : SET [0:1] OF IfcRelConnectsStructuralActivity FOR
RelatedStructuralActivity;

```

```

END_ENTITY;

```

```

ENTITY IfcStructuralAction

```

```

    ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralCurveAction, IfcStructuralPointAction,
IfcStructuralSurfaceAction))

```

```

    SUBTYPE OF(IfcStructuralActivity);

```

```

    DestabilizingLoad : OPTIONAL IfcBoolean;

```

```

END_ENTITY;

```

```

ENTITY IfcStructuralCurveAction

```

```

    SUBTYPE OF(IfcStructuralAction);

```

```

    ProjectedOrTrue : OPTIONAL IfcProjectedOrTrueLengthEnum;

```

```

    PredefinedType : IfcStructuralCurveActivityTypeEnum;

```

```

    WHERE

```

```

        PROJECTEDISGLOBAL : (NOT EXISTS(ProjectedOrTrue)) OR
        ((ProjectedOrTrue <> PROJECTED_LENGTH) OR

```



```

                (SELFWIfcStructuralActivity.GlobalOrLocal = GLOBAL_COORDS));
    HASOBJECTTYPE      : (PredefinedType <> IfcStructuralCurveActivityTypeEnum.USERDEFINED)
OR EXISTS(SELFWIfcObject.ObjectType);
    SUITABLEPREDEFINEDTYPE : PredefinedType <> IfcStructuralCurveActivityTypeEnum.EQUIDISTANT;
END_ENTITY;

```

```

ENTITY IfcStructuralLinearAction

```

```

    SUBTYPE OF(IfcStructuralCurveAction);

```

```

    WHERE

```

```

        SUITABLELOADTYPE      :      SIZEOF(['IFC4.IFCSTRUCTURALLOADLINEARFORCE',
'IFC4.IFCSTRUCTURALLOADTEMPERATURE'] * TYPEOF(SELFWIfcStructuralActivity.AppliedLoad)) = 1;

```

```

        CONSTPREDEFINEDTYPE    :      SELFWIfcStructuralCurveAction.PredefinedType      =
IfcStructuralCurveActivityTypeEnum.CONST;

```

```

END_ENTITY;

```

```

ENTITY IfcStructuralPointAction

```

```

    SUBTYPE OF(IfcStructuralAction);

```

```

    WHERE

```

```

        SUITABLELOADTYPE      :      SIZEOF(['IFC4.IFCSTRUCTURALLOADSINGLEFORCE',
'IFC4.IFCSTRUCTURALLOADSINGLEDISPLACEMENT'] * TYPEOF(SELFWIfcStructuralActivity.AppliedLoad)) = 1;

```

```

END_ENTITY;

```

```

ENTITY IfcStructuralSurfaceAction

```

```

    SUBTYPE OF(IfcStructuralAction);

```

```

        ProjectedOrTrue : OPTIONAL IfcProjectedOrTrueLengthEnum;

```

```

        PredefinedType  : IfcStructuralSurfaceActivityTypeEnum;

```

```

    WHERE

```

```

        PROJECTEDISGLOBAL : (NOT EXISTS(ProjectedOrTrue)) OR
                                ((ProjectedOrTrue <> PROJECTED_LENGTH) OR
                                (SELFWIfcStructuralActivity.GlobalOrLocal = GLOBAL_COORDS));

```

```

        HASOBJECTTYPE    : (PredefinedType <> IfcStructuralSurfaceActivityTypeEnum.USERDEFINED) OR

```

EXISTS(SELFWIfcObject.ObjectType);

END_ENTITY;

ENTITY IfcStructuralPlanarAction

SUBTYPE OF(IfcStructuralSurfaceAction);

WHERE

SUITABLELOADTYPE : SIZEOF(['IFC4.IFCSTRUCTURALLOADPLANARFORCE',
'IFC4.IFCSTRUCTURALLOADTEMPERATURE'] * TYPEOF(SELFWIfcStructuralActivity.AppliedLoad)) = 1;

CONSTPREDEFINEDTYPE : SELFWIfcStructuralSurfaceAction.PredefinedType =
IfcStructuralSurfaceActivityTypeEnum.CONST;

END_ENTITY;

ENTITY IfcStructuralReaction

ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralCurveReaction, IfcStructuralPointReaction,
IfcStructuralSurfaceReaction))

SUBTYPE OF(IfcStructuralActivity);

END_ENTITY;

ENTITY IfcStructuralCurveReaction

SUBTYPE OF(IfcStructuralReaction);

PredefinedType : IfcStructuralCurveActivityTypeEnum;

WHERE

HASOBJECTTYPE : (PredefinedType <> IfcStructuralCurveActivityTypeEnum.USERDEFINED)
OR EXISTS(SELFWIfcObject.ObjectType);

SUITABLEPREDEFINEDTYPE : (PredefinedType <> IfcStructuralCurveActivityTypeEnum.SINUS) AND
(PredefinedType <> IfcStructuralCurveActivityTypeEnum.PARABOLA);

END_ENTITY;

ENTITY IfcStructuralPointReaction

SUBTYPE OF(IfcStructuralReaction);

WHERE

```
SUITABLELOADTYPE          :          SIZEOF(['IFC4.IFCSTRUCTURALLOADSINGLEFORCE',
'IFC4.IFCSTRUCTURALLOADSINGLEDISPLACEMENT'] * TYPEOF(SELFWIfcStructuralActivity.AppliedLoad)) = 1;
END_ENTITY;
```

```
ENTITY IfcStructuralSurfaceReaction
```

```
  SUBTYPE OF(IfcStructuralReaction);
```

```
  PredefinedType : IfcStructuralSurfaceActivityTypeEnum;
```

```
  WHERE
```

```
    HASPREDEFINEDTYPE : (PredefinedType <> IfcStructuralSurfaceActivityTypeEnum.USERDEFINED) OR
    EXISTS(SELFWIfcObject.ObjectType);
```

```
END_ENTITY;
```

```
ENTITY IfcStructuralLoad
```

```
  ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralLoadConfiguration, IfcStructuralLoadOrResult));
```

```
  Name : OPTIONAL IfcLabel;
```

```
END_ENTITY;
```

```
ENTITY IfcStructuralLoadConfiguration
```

```
  SUBTYPE OF(IfcStructuralLoad);
```

```
  Values      : LIST [1:?] OF IfcStructuralLoadOrResult;
```

```
  Locations   : OPTIONAL LIST [1:?] OF LIST [1:2] OF IfcLengthMeasure;
```

```
  WHERE
```

```
    VALIDLISTSIZE : NOT EXISTS(Locations) OR (SIZEOF(Locations) = SIZEOF(Values));
```

```
END_ENTITY;
```

```
ENTITY IfcStructuralLoadOrResult
```

```
  ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralLoadStatic, IfcSurfaceReinforcementArea))
```

```
  SUBTYPE OF(IfcStructuralLoad);
```

```
END_ENTITY;
```

ENTITY IfcStructuralLoadStatic

ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralLoadLinearForce, IfcStructuralLoadPlanarForce, IfcStructuralLoadSingleDisplacement, IfcStructuralLoadSingleForce, IfcStructuralLoadTemperature))

SUBTYPE OF(IfcStructuralLoadOrResult);

END_ENTITY;

ENTITY IfcStructuralLoadLinearForce

SUBTYPE OF(IfcStructuralLoadStatic);

LinearForceX : OPTIONAL IfcLinearForceMeasure;

LinearForceY : OPTIONAL IfcLinearForceMeasure;

LinearForceZ : OPTIONAL IfcLinearForceMeasure;

LinearMomentX : OPTIONAL IfcLinearMomentMeasure;

LinearMomentY : OPTIONAL IfcLinearMomentMeasure;

LinearMomentZ : OPTIONAL IfcLinearMomentMeasure;

END_ENTITY;

ENTITY IfcStructuralLoadPlanarForce

SUBTYPE OF(IfcStructuralLoadStatic);

PlanarForceX : OPTIONAL IfcPlanarForceMeasure;

PlanarForceY : OPTIONAL IfcPlanarForceMeasure;

PlanarForceZ : OPTIONAL IfcPlanarForceMeasure;

END_ENTITY;

ENTITY IfcStructuralLoadSingleDisplacement

SUBTYPE OF(IfcStructuralLoadStatic);

DisplacementX : OPTIONAL IfcLengthMeasure;

DisplacementY : OPTIONAL IfcLengthMeasure;

DisplacementZ : OPTIONAL IfcLengthMeasure;

RotationalDisplacementRX : OPTIONAL IfcPlaneAngleMeasure;

RotationalDisplacementRY : OPTIONAL IfcPlaneAngleMeasure;

```
    RotationalDisplacementRZ : OPTIONAL IfcPlaneAngleMeasure;  
END_ENTITY;
```

```
ENTITY IfcStructuralLoadSingleDisplacementDistortion  
    SUBTYPE OF(IfcStructuralLoadSingleDisplacement);  
    Distortion : OPTIONAL IfcCurvatureMeasure;  
END_ENTITY;
```

```
ENTITY IfcStructuralLoadSingleForce  
    SUBTYPE OF(IfcStructuralLoadStatic);  
    ForceX : OPTIONAL IfcForceMeasure;  
    ForceY : OPTIONAL IfcForceMeasure;  
    ForceZ : OPTIONAL IfcForceMeasure;  
    MomentX : OPTIONAL IfcTorqueMeasure;  
    MomentY : OPTIONAL IfcTorqueMeasure;  
    MomentZ : OPTIONAL IfcTorqueMeasure;  
END_ENTITY;
```

```
ENTITY IfcStructuralLoadSingleForceWarping  
    SUBTYPE OF(IfcStructuralLoadSingleForce);  
    WarpingMoment : OPTIONAL IfcWarpingMomentMeasure;  
END_ENTITY;
```

```
ENTITY IfcStructuralLoadTemperature  
    SUBTYPE OF(IfcStructuralLoadStatic);  
    DeltaTConstant : OPTIONAL IfcThermodynamicTemperatureMeasure;  
    DeltaTY : OPTIONAL IfcThermodynamicTemperatureMeasure;  
    DeltaTZ : OPTIONAL IfcThermodynamicTemperatureMeasure;  
END_ENTITY;
```

ENTITY IfcSurfaceReinforcementArea

SUBTYPE OF(IfcStructuralLoadOrResult);

SurfaceReinforcement1 : OPTIONAL LIST [2:3] OF IfcLengthMeasure;

SurfaceReinforcement2 : OPTIONAL LIST [2:3] OF IfcLengthMeasure;

ShearReinforcement : OPTIONAL IfcRatioMeasure;

WHERE

SURFACEANDORSHEARAREASPECIFIED : EXISTS(SurfaceReinforcement1) OR
EXISTS(SurfaceReinforcement2) OR EXISTS(ShearReinforcement);

NONNEGATIVEAREA1 : (NOT EXISTS(SurfaceReinforcement1)) OR (
(SurfaceReinforcement1[1] >= 0.) AND
(SurfaceReinforcement1[2] >= 0.) AND
((SIZEOF(SurfaceReinforcement1) = 1) OR
(SurfaceReinforcement1[1] >= 0.))
);

NONNEGATIVEAREA2 : (NOT EXISTS(SurfaceReinforcement2)) OR (
(SurfaceReinforcement2[1] >= 0.) AND
(SurfaceReinforcement2[2] >= 0.) AND
((SIZEOF(SurfaceReinforcement2) = 1) OR
(SurfaceReinforcement2[1] >= 0.))
);

NONNEGATIVEAREA3 : (NOT EXISTS(ShearReinforcement)) OR (ShearReinforcement >= 0.);

END_ENTITY;

ENTITY IfcRelConnectsStructuralActivity

SUBTYPE OF(IfcRelConnects);

RelatingElement : IfcStructuralActivityAssignmentSelect;

RelatedStructuralActivity : IfcStructuralActivity;

END_ENTITY;

ENTITY IfcStructuralItem

```

ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralConnection, IfcStructuralMember))
SUBTYPE OF(IfcProduct);
INVERSE
    AssignedStructuralActivity : SET OF IfcRelConnectsStructuralActivity FOR RelatingElement;
END_ENTITY;

```

```

ENTITY IfcStructuralConnection
    ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralCurveConnection, IfcStructuralPointConnection,
IfcStructuralSurfaceConnection))
    SUBTYPE OF(IfcStructuralItem);
    AppliedCondition : OPTIONAL IfcBoundaryCondition;
INVERSE
    ConnectsStructuralMembers : SET [1:?] OF IfcRelConnectsStructuralMember FOR
RelatedStructuralConnection;
END_ENTITY;

```

```

ENTITY IfcStructuralCurveConnection
    SUBTYPE OF(IfcStructuralConnection);
    Axis : IfcDirection;
END_ENTITY;

```

```

ENTITY IfcStructuralPointConnection
    SUBTYPE OF(IfcStructuralConnection);
    ConditionCoordinateSystem : OPTIONAL IfcAxis2Placement3D;
END_ENTITY;

```

```

ENTITY IfcStructuralSurfaceConnection
    SUBTYPE OF(IfcStructuralConnection);
END_ENTITY;

```

```

ENTITY IfcBoundaryCondition
  ABSTRACT SUPERTYPE OF (ONEOF(IfcBoundaryEdgeCondition, IfcBoundaryFaceCondition,
IfcBoundaryNodeCondition));
  Name : OPTIONAL IfcLabel;
END_ENTITY;

```

```

ENTITY IfcBoundaryEdgeCondition
  SUBTYPE OF(IfcBoundaryCondition);
  TranslationalStiffnessByLengthX : OPTIONAL IfcModulusOfTranslationalSubgradeReactionSelect;
  TranslationalStiffnessByLengthY : OPTIONAL IfcModulusOfTranslationalSubgradeReactionSelect;
  TranslationalStiffnessByLengthZ : OPTIONAL IfcModulusOfTranslationalSubgradeReactionSelect;
  RotationalStiffnessByLengthX : OPTIONAL IfcModulusOfRotationalSubgradeReactionSelect;
  RotationalStiffnessByLengthY : OPTIONAL IfcModulusOfRotationalSubgradeReactionSelect;
  RotationalStiffnessByLengthZ : OPTIONAL IfcModulusOfRotationalSubgradeReactionSelect;
END_ENTITY;

```

```

ENTITY IfcBoundaryFaceCondition
  SUBTYPE OF(IfcBoundaryCondition);
  TranslationalStiffnessByAreaX : OPTIONAL IfcModulusOfSubgradeReactionSelect;
  TranslationalStiffnessByAreaY : OPTIONAL IfcModulusOfSubgradeReactionSelect;
  TranslationalStiffnessByAreaZ : OPTIONAL IfcModulusOfSubgradeReactionSelect;
END_ENTITY;

```

```

ENTITY IfcBoundaryNodeCondition
  SUBTYPE OF(IfcBoundaryCondition);
  TranslationalStiffnessX : OPTIONAL IfcTranslationalStiffnessSelect;
  TranslationalStiffnessY : OPTIONAL IfcTranslationalStiffnessSelect;
  TranslationalStiffnessZ : OPTIONAL IfcTranslationalStiffnessSelect;
  RotationalStiffnessX : OPTIONAL IfcRotationalStiffnessSelect;
  RotationalStiffnessY : OPTIONAL IfcRotationalStiffnessSelect;

```



```
    RotationalStiffnessZ      : OPTIONAL IfcRotationalStiffnessSelect;  
END_ENTITY;
```

```
ENTITY IfcBoundaryNodeConditionWarping  
    SUBTYPE OF(IfcBoundaryNodeCondition);  
    WarpingStiffness : OPTIONAL IfcWarpingStiffnessSelect;  
END_ENTITY;
```

```
ENTITY IfcRelConnectsStructuralMember  
    SUBTYPE OF(IfcRelConnects);  
    RelatingStructuralMember      : IfcStructuralMember;  
    RelatedStructuralConnection : IfcStructuralConnection;  
    AppliedCondition              : OPTIONAL IfcBoundaryCondition;  
    AdditionalConditions          : OPTIONAL IfcStructuralConnectionCondition;  
    SupportedLength              : OPTIONAL IfcLengthMeasure;  
    ConditionCoordinateSystem     : OPTIONAL IfcAxis2Placement3D;  
END_ENTITY;
```

```
ENTITY IfcRelConnectsWithEccentricity  
    SUBTYPE OF(IfcRelConnectsStructuralMember);  
    ConnectionConstraint : IfcConnectionGeometry;  
END_ENTITY;
```

```
ENTITY IfcStructuralMember  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcStructuralCurveMember, IfcStructuralSurfaceMember))  
    SUBTYPE OF(IfcStructuralItem);  
    INVERSE  
    ConnectedBy : SET OF IfcRelConnectsStructuralMember FOR RelatingStructuralMember;  
END_ENTITY;
```

```

ENTITY IfcStructuralCurveMember
  SUBTYPE OF(IfcStructuralMember);
  PredefinedType : IfcStructuralCurveMemberTypeEnum;
  Axis           : IfcDirection;
  WHERE
    HASOBJECTTYPE : (PredefinedType <> IfcStructuralCurveMemberTypeEnum.USERDEFINED) OR
  EXISTS(SELFWIfcObject.ObjectType);
END_ENTITY;

ENTITY IfcStructuralCurveMemberVarying
  SUBTYPE OF(IfcStructuralCurveMember);
END_ENTITY;

ENTITY IfcStructuralSurfaceMember
  SUBTYPE OF(IfcStructuralMember);
  PredefinedType : IfcStructuralSurfaceMemberTypeEnum;
  Thickness      : OPTIONAL IfcPositiveLengthMeasure;
  WHERE
    HASOBJECTTYPE : (PredefinedType <> IfcStructuralSurfaceMemberTypeEnum.USERDEFINED) OR
  EXISTS(SELFWIfcObject.ObjectType);
END_ENTITY;

ENTITY IfcStructuralSurfaceMemberVarying
  SUBTYPE OF(IfcStructuralSurfaceMember);
END_ENTITY;

ENTITY IfcStructuralConnectionCondition
  ABSTRACT SUPERTYPE OF (ONEOF(IfcFailureConnectionCondition, IfcSlippageConnectionCondition));
  Name : OPTIONAL IfcLabel;
END_ENTITY;

```

```
ENTITY IfcFailureConnectionCondition
  SUBTYPE OF(IfcStructuralConnectionCondition);
  TensionFailureX : OPTIONAL IfcForceMeasure;
  TensionFailureY : OPTIONAL IfcForceMeasure;
  TensionFailureZ : OPTIONAL IfcForceMeasure;
  CompressionFailureX : OPTIONAL IfcForceMeasure;
  CompressionFailureY : OPTIONAL IfcForceMeasure;
  CompressionFailureZ : OPTIONAL IfcForceMeasure;
END_ENTITY;
```

```
ENTITY IfcSlippageConnectionCondition
  SUBTYPE OF(IfcStructuralConnectionCondition);
  SlippageX : OPTIONAL IfcLengthMeasure;
  SlippageY : OPTIONAL IfcLengthMeasure;
  SlippageZ : OPTIONAL IfcLengthMeasure;
END_ENTITY;
```

```
ENTITY IfcPositioningElement
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcProduct);
END_ENTITY;
```

```
ENTITY IfcAlignment
  SUBTYPE OF(IfcPositioningElement);
  PredefinedType : OPTIONAL IfcAlignmentTypeEnum;
  Horizontal : OPTIONAL IfcAlignment2DHorizontal;
  Vertical : OPTIONAL IfcAlignment2DVertical;
  LinearRefMethod : OPTIONAL IfcLabel;
WHERE
```

VALIDCOMBINATION : (EXISTS(Horizontal) AND EXISTS(Vertical)) OR
(EXISTS(Horizontal) AND NOT(EXISTS(Vertical))) OR
(NOT(EXISTS(Horizontal)) AND NOT(EXISTS(Vertical)));

END_ENTITY;

ENTITY IfcAlignment2DHorizontal;

Segments : LIST [1:?] OF IfcAlignment2DHorizontalSegment;

StartDistAlong : OPTIONAL IfcLengthMeasure;

INVERSE

ToAlignment : SET [1:?] OF IfcAlignment FOR Horizontal;

END_ENTITY;

ENTITY IfcAlignment2DHorizontalSegment

SUBTYPE OF(IfcAlignment2DSegment);

CurveGeometry : IfcCurveSegment2D;

INVERSE

ToHorizontal : SET [1:1] OF IfcAlignment2DHorizontal FOR Segments;

END_ENTITY;

ENTITY IfcAlignment2DSegment

ABSTRACT SUPERTYPE OF (ONEOF(IfcAlignment2DHorizontalSegment, IfcAlignment2DVerticalSegment));

TangentialContinuity : OPTIONAL IfcBoolean;

EndTag : OPTIONAL IfcLabel;

StartTag : OPTIONAL IfcLabel;

END_ENTITY;

ENTITY IfcAlignment2DVerticalSegment

ABSTRACT SUPERTYPE OF (ONEOF(IfcAlignment2DVerSegCircularArc, IfcAlignment2DVerSegLine,
IfcAlignment2DVerSegParabolicArc))

SUBTYPE OF(IfcAlignment2DSegment);

```

    StartHeight      : IfcLengthMeasure;
    HorizontalLength : IfcPositiveLengthMeasure;
    StartGradient    : IfcRatioMeasure;
    StartDistAlong   : IfcLengthMeasure;
    INVERSE
    ToVertical        : SET [1:1] OF IfcAlignment2DVertical FOR Segments;
END_ENTITY;

```

```

ENTITY IfcAlignment2DVerSegCircularArc
    SUBTYPE OF(IfcAlignment2DVerticalSegment);
    IsConvex : IfcBoolean;
    Radius    : IfcPositiveLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcAlignment2DVerSegLine
    SUBTYPE OF(IfcAlignment2DVerticalSegment);
END_ENTITY;

```

```

ENTITY IfcAlignment2DVerSegParabolicArc
    SUBTYPE OF(IfcAlignment2DVerticalSegment);
    IsConvex      : IfcBoolean;
    ParabolaConstant : IfcPositiveLengthMeasure;
END_ENTITY;

```

```

ENTITY IfcAlignment2DVertical;
    Segments : LIST [1:?] OF IfcAlignment2DVerticalSegment;
    INVERSE
    ToAlignment : SET [1:1] OF IfcAlignment FOR Vertical;
END_ENTITY;

```

```

ENTITY IfcCurveSegment2D
  ABSTRACT SUPERTYPE OF (ONEOF(IfcCircularArcSegment2D, IfcClothoidalArcSegment2D,
IfcLineSegment2D));
  SegmentLength : IfcPositiveLengthMeasure;
  StartDirection : IfcPlaneAngleMeasure;
  StartPoint : IfcCartesianPoint;
END_ENTITY;

```

```

ENTITY IfcCircularArcSegment2D
  SUBTYPE OF(IfcCurveSegment2D);
  Radius : IfcPositiveLengthMeasure;
  IsCCW : IfcBoolean;
END_ENTITY;

```

```

ENTITY IfcClothoidalArcSegment2D
  SUBTYPE OF(IfcCurveSegment2D);
  StartRadius : OPTIONAL IfcPositiveLengthMeasure;
  IsCCW : IfcBoolean;
  IsEntry : IfcBoolean;
  ClothoidConstant : IfcReal;
END_ENTITY;

```

```

ENTITY IfcLineSegment2D
  SUBTYPE OF(IfcCurveSegment2D);
END_ENTITY;

```

```

ENTITY IfcObjectPlacement
  ABSTRACT SUPERTYPE OF (ONEOF(IfcGridPlacement, IfcLocalPlacement));
  INVERSE
  PlacesObject : SET OF IfcProduct FOR ObjectPlacement;

```

```
    ReferencedByPlacements : SET OF IfcLocalPlacement FOR PlacementRelTo;
END_ENTITY;
```

```
ENTITY IfcGridPlacement
  SUBTYPE OF(IfcObjectPlacement);
  PlacementLocation      : IfcVirtualGridIntersection;
  PlacementRefDirection : OPTIONAL IfcGridPlacementDirectionSelect;
END_ENTITY;
```

```
ENTITY IfcLocalPlacement
  SUBTYPE OF(IfcObjectPlacement);
  PlacementRelTo      : OPTIONAL IfcObjectPlacement;
  RelativePlacement   : IfcAxis2Placement;
  WHERE
    WR21 : IfcCorrectLocalPlacement(RelativePlacement, PlacementRelTo);
END_ENTITY;
```

```
ENTITY IfcRelAssignsToProduct
  SUBTYPE OF(IfcRelAssigns);
  RelatingProduct : IfcProductSelect;
  WHERE
    NOSELFREFERENCE : SIZEOF(QUERY(Temp < * SELFW IfcRelAssigns.RelatedObjects |
    RelatingProduct :=: Temp)) = 0;
END_ENTITY;
```

```
ENTITY IfcTypeProduct
  SUPERTYPE OF (ONEOF(IfcDoorStyle, IfcElementType, IfcSpatialElementType, IfcWindowStyle))
  SUBTYPE OF(IfcTypeObject);
  RepresentationMaps : OPTIONAL LIST [1:?] OF IfcRepresentationMap;
  Tag                 : OPTIONAL IfcLabel;
```

INVERSE

ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;

WHERE

APPLICABLEOCCURRENCE : NOT(EXISTS(SELFWIfcTypeObject.Types[1])) OR
(SIZEOF(QUERY(temp <* SELFWIfcTypeObject.Types[1].RelatedObjects |
NOT('IFC4.IFCPRODUCT' IN TYPEOF(temp)))
) = 0);

END_ENTITY;

ENTITY IfcTypeObject

SUPERTYPE OF (ONEOF(IfcTypeProcess, IfcTypeProduct, IfcTypeResource))

SUBTYPE OF(IfcObjectDefinition);

ApplicableOccurrence : OPTIONAL IfcIdentifier;

HasPropertySets : OPTIONAL SET [1:?] OF IfcPropertySetDefinition;

INVERSE

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

WHERE

NAMEREQUIRED : EXISTS(SELFWIfcRoot.Name);

UNIQUEPROPERTYSETNAMES : (NOT(EXISTS(HasPropertySets))) OR
IfcUniquePropertySetNames(HasPropertySets);

END_ENTITY;

ENTITY IfcTypeProcess

ABSTRACT SUPERTYPE OF (ONEOF(IfcEventType, IfcProcedureType, IfcTaskType))

SUBTYPE OF(IfcTypeObject);

Identification : OPTIONAL IfcIdentifier;

LongDescription : OPTIONAL IfcText;

ProcessType : OPTIONAL IfcLabel;

INVERSE

OperatesOn : SET OF IfcRelAssignsToProcess FOR RelatingProcess;

END_ENTITY;

ENTITY IfcEventType

SUBTYPE OF(IfcTypeProcess);

PredefinedType : IfcEventTypeEnum;

EventTriggerType : IfcEventTriggerTypeEnum;

UserDefinedEventTriggerType : OPTIONAL IfcLabel;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcEventTypeEnum.USERDEFINED) OR
((PredefinedType = IfcEventTypeEnum.USERDEFINED) AND EXISTS(SELFWIfcTypeProcess.ProcessType));

CORRECTEVENTTRIGGERTYPE : (EventTriggerType <> IfcEventTriggerTypeEnum.USERDEFINED) OR
((EventTriggerType = IfcEventTriggerTypeEnum.USERDEFINED) AND
EXISTS(UserDefinedEventTriggerType));

END_ENTITY;

ENTITY IfcProcedureType

SUBTYPE OF(IfcTypeProcess);

PredefinedType : IfcProcedureTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcProcedureTypeEnum.USERDEFINED) OR
((PredefinedType = IfcProcedureTypeEnum.USERDEFINED) AND EXISTS(SELFWIfcTypeProcess.ProcessType));

END_ENTITY;

ENTITY IfcTaskType

SUBTYPE OF(IfcTypeProcess);

PredefinedType : IfcTaskTypeEnum;

WorkMethod : OPTIONAL IfcLabel;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTaskTypeEnum.USERDEFINED) OR ((PredefinedType
= IfcTaskTypeEnum.USERDEFINED) AND EXISTS(SELFWIfcTypeProcess.ProcessType));

END_ENTITY;

```

ENTITY IfcRelAssignsToProcess
  SUBTYPE OF(IfcRelAssigns);
  RelatingProcess : IfcProcessSelect;
  QuantityInProcess : OPTIONAL IfcMeasureWithUnit;
  WHERE
    NOSELFREFERENCE : SIZEOF(QUERY(Temp <* SELFWIfcRelAssigns.RelatedObjects |
    RelatingProcess :=: Temp)) = 0;
END_ENTITY;

```

```

ENTITY IfcProcess
  ABSTRACT SUPERTYPE OF (ONEOF(IfcEvent, IfcProcedure, IfcTask))
  SUBTYPE OF(IfcObject);
  Identification : OPTIONAL IfcIdentifier;
  LongDescription : OPTIONAL IfcText;
  INVERSE
    IsPredecessorTo : SET OF IfcRelSequence FOR RelatingProcess;
    IsSuccessorFrom : SET OF IfcRelSequence FOR RelatedProcess;
    OperatesOn : SET OF IfcRelAssignsToProcess FOR RelatingProcess;
END_ENTITY;

```

```

ENTITY IfcEvent
  SUBTYPE OF(IfcProcess);
  PredefinedType : OPTIONAL IfcEventTypeEnum;
  EventTriggerType : OPTIONAL IfcEventTriggerTypeEnum;
  UserDefinedEventTriggerType : OPTIONAL IfcLabel;
  EventOccurrenceTime : OPTIONAL IfcEventTime;
  WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
    IfcEventTypeEnum.USERDEFINED) OR ((PredefinedType = IfcEventTypeEnum.USERDEFINED) AND
    EXISTS(SELFWIfcObject.ObjectType));

```

```
CORRECTTYPEASSIGNED      : NOT(EXISTS(EventTriggerType)) OR (EventTriggerType <>
IfcEventTriggerTypeEnum.USERDEFINED) OR ((EventTriggerType = IfcEventTriggerTypeEnum.USERDEFINED)
AND EXISTS(UserDefinedEventTriggerType));
```

```
END_ENTITY;
```

```
ENTITY IfcProcedure
```

```
  SUBTYPE OF(IfcProcess);
```

```
  PredefinedType : OPTIONAL IfcProcedureTypeEnum;
```

```
  WHERE
```

```
    HASNAME          : EXISTS(SELFWIfcRoot.Name);
```

```
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcProcedureTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcProcedureTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcObject.ObjectType));
```

```
END_ENTITY;
```

```
ENTITY IfcTask
```

```
  SUBTYPE OF(IfcProcess);
```

```
  Status          : OPTIONAL IfcLabel;
```

```
  WorkMethod      : OPTIONAL IfcLabel;
```

```
  IsMilestone     : IfcBoolean;
```

```
  Priority         : OPTIONAL IfcInteger;
```

```
  TaskTime        : OPTIONAL IfcTaskTime;
```

```
  PredefinedType : OPTIONAL IfcTaskTypeEnum;
```

```
  WHERE
```

```
    HASNAME          : EXISTS(SELFWIfcRoot.Name);
```

```
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR (PredefinedType <>
IfcTaskTypeEnum.USERDEFINED) OR ((PredefinedType = IfcTaskTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcObject.ObjectType));
```

```
END_ENTITY;
```

```
ENTITY IfcRelSequence
```

SUBTYPE OF(IfcRelConnects);

RelatingProcess : IfcProcess;
RelatedProcess : IfcProcess;
TimeLag : OPTIONAL IfcLagTime;
SequenceType : OPTIONAL IfcSequenceEnum;
UserDefinedSequenceType : OPTIONAL IfcLabel;

WHERE

AVOIDINCONSISTENTSEQUENCE : RelatingProcess :<>: RelatedProcess;

CORRECTSEQUENCETYPE : (SequenceType <> IfcSequenceEnum.USERDEFINED) OR ((SequenceType = IfcSequenceEnum.USERDEFINED) AND EXISTS(UserDefinedSequenceType));

END_ENTITY;

ENTITY IfcTypeResource

ABSTRACT SUPERTYPE

SUBTYPE OF(IfcTypeObject);

Identification : OPTIONAL IfcIdentifier;
LongDescription : OPTIONAL IfcText;
ResourceType : OPTIONAL IfcLabel;

INVERSE

ResourceOf : SET OF IfcRelAssignsToResource FOR RelatingResource;

END_ENTITY;

ENTITY IfcConstructionResourceType

ABSTRACT SUPERTYPE OF (ONEOF(IfcConstructionEquipmentResourceType,
IfcConstructionMaterialResourceType ANDOR IfcEarthworkMaterialResourceType_K,
IfcConstructionProductResourceType, IfcCrewResourceType, IfcLaborResourceType,
IfcSubContractResourceType))

SUBTYPE OF(IfcTypeResource);

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

END_ENTITY;

ENTITY IfcConstructionEquipmentResourceType

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcConstructionEquipmentResourceTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <>
IfcConstructionEquipmentResourceTypeEnum.USERDEFINED) OR
((PredefinedType =
IfcConstructionEquipmentResourceTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcTypeResource.ResourceType));

END_ENTITY;

ENTITY IfcConstructionMaterialResourceType

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcConstructionMaterialResourceTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <>
IfcConstructionMaterialResourceTypeEnum.USERDEFINED) OR
((PredefinedType =
IfcConstructionMaterialResourceTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcTypeResource.ResourceType));

END_ENTITY;

ENTITY IfcEarthworkMaterialResourceType_K

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcEarthworkMaterialResourceTypeEnum_K;

END_ENTITY;

ENTITY IfcConstructionProductResourceType

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcConstructionProductResourceTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcConstructionProductResourceTypeEnum.USERDEFINED) OR
((PredefinedType = IfcConstructionProductResourceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcTypeResource.ResourceType));

```

END_ENTITY;

ENTITY IfcCrewResourceType

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcCrewResourceTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCrewResourceTypeEnum.USERDEFINED) OR
((PredefinedType = IfcCrewResourceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcTypeResource.ResourceType));

```

END_ENTITY;

ENTITY IfcLaborResourceType

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcLaborResourceTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcLaborResourceTypeEnum.USERDEFINED) OR
((PredefinedType = IfcLaborResourceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcTypeResource.ResourceType));

```

END_ENTITY;

ENTITY IfcSubContractResourceType

SUBTYPE OF(IfcConstructionResourceType);

PredefinedType : IfcSubContractResourceTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSubContractResourceTypeEnum.USERDEFINED) OR
((PredefinedType = IfcSubContractResourceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcTypeResource.ResourceType));

```

END_ENTITY;

ENTITY IfcRelAssignsToResource

SUBTYPE OF(IfcRelAssigns);

RelatingResource : IfcResourceSelect;

WHERE

NOSELFREFERENCE : SIZEOF(QUERY(Temp < * SELFIfcRelAssigns.RelatedObjects |
RelatingResource := Temp)) = 0;

END_ENTITY;

ENTITY IfcResource

ABSTRACT SUPERTYPE

SUBTYPE OF(IfcObject);

Identification : OPTIONAL IfcIdentifier;

LongDescription : OPTIONAL IfcText;

INVERSE

ResourceOf : SET OF IfcRelAssignsToResource FOR RelatingResource;

END_ENTITY;

ENTITY IfcConstructionResource

ABSTRACT SUPERTYPE OF (ONEOF(IfcConstructionEquipmentResource,
IfcConstructionMaterialResource, IfcConstructionProductResource, IfcCrewResource,
IfcLaborResource, IfcSubContractResource, IfcEarthworkMaterialResource_K))

SUBTYPE OF(IfcResource);

Usage : OPTIONAL IfcResourceTime;

BaseCosts : OPTIONAL LIST [1:?] OF IfcAppliedValue;

BaseQuantity : OPTIONAL IfcPhysicalQuantity;

END_ENTITY;

ENTITY IfcConstructionEquipmentResource

SUBTYPE OF(IfcConstructionResource);

```

    PredefinedType : OPTIONAL IfcConstructionEquipmentResourceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
        (PredefinedType
IfcConstructionEquipmentResourceTypeEnum.USERDEFINED) OR
        ((PredefinedType
IfcConstructionEquipmentResourceTypeEnum.USERDEFINED) AND EXISTS (SELFIfcObject.ObjectType));
END_ENTITY;

```

```

ENTITY IfcConstructionMaterialResource
    SUBTYPE OF(IfcConstructionResource);
    PredefinedType : OPTIONAL IfcConstructionMaterialResourceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
        (PredefinedType
IfcConstructionMaterialResourceTypeEnum.USERDEFINED) OR
        ((PredefinedType
IfcConstructionMaterialResourceTypeEnum.USERDEFINED) AND EXISTS (SELFIfcObject.ObjectType));
END_ENTITY;

```

```

ENTITY IfcConstructionProductResource
    SUBTYPE OF(IfcConstructionResource);
    PredefinedType : OPTIONAL IfcConstructionProductResourceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
        (PredefinedType
IfcConstructionProductResourceTypeEnum.USERDEFINED) OR
        ((PredefinedType
IfcConstructionProductResourceTypeEnum.USERDEFINED) AND EXISTS (SELFIfcObject.ObjectType));
END_ENTITY;

```

```

ENTITY IfcCrewResource

```



```

SUBTYPE OF(IfcConstructionResource);
    PredefinedType : OPTIONAL IfcCrewResourceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
        (PredefinedType <> IfcCrewResourceTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcCrewResourceTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
END_ENTITY;

```

```

ENTITY IfcLaborResource
SUBTYPE OF(IfcConstructionResource);
    PredefinedType : OPTIONAL IfcLaborResourceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
        (PredefinedType <> IfcLaborResourceTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcLaborResourceTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
END_ENTITY;

```

```

ENTITY IfcSubContractResource
SUBTYPE OF(IfcConstructionResource);
    PredefinedType : OPTIONAL IfcSubContractResourceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
        (PredefinedType <> IfcSubContractResourceTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcSubContractResourceTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
END_ENTITY;

```

```

ENTITY IfcEarthworkMaterialResource_K
SUBTYPE OF(IfcConstructionResource);

```

PredefinedType : OPTIONAL IfcEarthworkMaterialResourceTypeEnum_K;
END_ENTITY;

ENTITY IfcReIDefinesByType
SUBTYPE OF(IfcReIDefines);
RelatedObjects : SET [1:?] OF IfcObject;
RelatingType : IfcTypeObject;
END_ENTITY;

ENTITY IfcReIDefines
ABSTRACT SUPERTYPE OF (ONEOF(IfcReIDefinesByObject, IfcReIDefinesByProperties,
IfcReIDefinesByTemplate, IfcReIDefinesByType))
SUBTYPE OF(IfcRelationship);
END_ENTITY;

ENTITY IfcReIDefinesByObject
SUBTYPE OF(IfcReIDefines);
RelatedObjects : SET [1:?] OF IfcObject;
RelatingObject : IfcObject;
END_ENTITY;

ENTITY IfcReIDefinesByProperties
SUBTYPE OF(IfcReIDefines);
RelatedObjects : SET [1:?] OF IfcObjectDefinition;
RelatingPropertyDefinition : IfcPropertySetDefinitionSelect;
END_ENTITY;

ENTITY IfcReIDefinesByTemplate
SUBTYPE OF(IfcReIDefines);
RelatedPropertySets : SET [1:?] OF IfcPropertySetDefinition;

```
    RelatingTemplate      : IfcPropertySetTemplate;  
END_ENTITY;
```

```
ENTITY IfcPropertySetTemplate
```

```
    SUBTYPE OF(IfcPropertyTemplateDefinition);
```

```
    TemplateType          : OPTIONAL IfcPropertySetTemplateTypeEnum;
```

```
    ApplicableEntity      : OPTIONAL IfcIdentifier;
```

```
    HasPropertyTemplates : SET [1:?] OF IfcPropertyTemplate;
```

```
INVERSE
```

```
    Defines                : SET OF IfcRelDefinesByTemplate FOR RelatingTemplate;
```

```
WHERE
```

```
    EXISTSNAME             : EXISTS(SELFIfcRoot.Name);
```

```
    UNIQUEPROPERTYNAMES : IfcUniquePropertyTemplateNameNames(HasPropertyTemplates);
```

```
END_ENTITY;
```

```
ENTITY IfcPropertyTemplateDefinition
```

```
    ABSTRACT SUPERTYPE OF (ONEOF(IfcPropertySetTemplate, IfcPropertyTemplate))
```

```
    SUBTYPE OF(IfcPropertyDefinition);
```

```
END_ENTITY;
```

```
ENTITY IfcPropertyTemplate
```

```
    ABSTRACT SUPERTYPE OF (ONEOF(IfcComplexPropertyTemplate, IfcSimplePropertyTemplate))
```

```
    SUBTYPE OF(IfcPropertyTemplateDefinition);
```

```
INVERSE
```

```
    PartOfComplexTemplate : SET OF IfcComplexPropertyTemplate FOR HasPropertyTemplates;
```

```
    PartOfPsetTemplate    : SET OF IfcPropertySetTemplate FOR HasPropertyTemplates;
```

```
END_ENTITY;
```

```
ENTITY IfcComplexPropertyTemplate
```

```
    SUBTYPE OF(IfcPropertyTemplate);
```

```

UsageName          : OPTIONAL IfcLabel;
TemplateType       : OPTIONAL IfcComplexPropertyTemplateTypeEnum;
HasPropertyTemplates : OPTIONAL SET [1:?] OF IfcPropertyTemplate;
WHERE
  UNIQUEPROPERTYNAMES : IfcUniquePropertyTemplateNameNames(HasPropertyTemplates);
  NOSELFREFERENCE     : SIZEOF(QUERY(temp <* HasPropertyTemplates | SELF :=: temp)) = 0;
END_ENTITY;

```

```

ENTITY IfcSimplePropertyTemplate
  SUBTYPE OF(IfcPropertyTemplate);
  TemplateType          : OPTIONAL IfcSimplePropertyTemplateTypeEnum;
  PrimaryMeasureType   : OPTIONAL IfcLabel;
  SecondaryMeasureType : OPTIONAL IfcLabel;
  Enumerators          : OPTIONAL IfcPropertyEnumeration;
  PrimaryUnit          : OPTIONAL IfcUnit;
  SecondaryUnit        : OPTIONAL IfcUnit;
  Expression           : OPTIONAL IfcLabel;
  AccessState          : OPTIONAL IfcStateEnum;
END_ENTITY;

```

```

ENTITY IfcPropertyEnumeration
  SUBTYPE OF(IfcPropertyAbstraction);
  Name          : IfcLabel;
  EnumerationValues : LIST [1:?] OF IfcValue;
  Unit          : OPTIONAL IfcUnit;
UNIQUE
  UR1 : Name;
WHERE
  WR01 : SIZEOF(QUERY(temp <* SELF.EnumerationValues |
    NOT(TYPEOF(SELF.EnumerationValues[1]) = TYPEOF(temp)))

```

)) = 0;

END_ENTITY;

ENTITY IfcPropertyAbstraction

ABSTRACT SUPERTYPE OF (ONEOF(IfcExtendedProperties, IfcPreDefinedProperties, IfcProperty, IfcPropertyEnumeration));

INVERSE

HasExternalReferences : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;

END_ENTITY;

ENTITY IfcExtendedProperties

ABSTRACT SUPERTYPE OF (ONEOF(IfcMaterialProperties, IfcProfileProperties))

SUBTYPE OF(IfcPropertyAbstraction);

Name : OPTIONAL IfcIdentifier;

Description : OPTIONAL IfcText;

Properties : SET [1:?] OF IfcProperty;

END_ENTITY;

ENTITY IfcMaterialProperties

SUBTYPE OF(IfcExtendedProperties);

Material : IfcMaterialDefinition;

END_ENTITY;

ENTITY IfcProfileProperties

SUBTYPE OF(IfcExtendedProperties);

ProfileDefinition : IfcProfileDef;

END_ENTITY;

ENTITY IfcProperty

ABSTRACT SUPERTYPE OF (ONEOF(IfcComplexProperty, IfcSimpleProperty))

SUBTYPE OF(IfcPropertyAbstraction);

Name : IfcIdentifier;

Description : OPTIONAL IfcText;

INVERSE

PartOfPset : SET OF IfcPropertySet FOR HasProperties;

PropertyForDependance : SET OF IfcPropertyDependencyRelationship FOR DependingProperty;

PropertyDependsOn : SET OF IfcPropertyDependencyRelationship FOR DependantProperty;

PartOfComplex : SET OF IfcComplexProperty FOR HasProperties;

HasConstraints : SET OF IfcResourceConstraintRelationship FOR RelatedResourceObjects;

HasApprovals : SET OF IfcResourceApprovalRelationship FOR RelatedResourceObjects;

END_ENTITY;

ENTITY IfcComplexProperty

SUBTYPE OF(IfcProperty);

UsageName : IfcIdentifier;

HasProperties : SET [1:?] OF IfcProperty;

WHERE

WR21 : SIZEOF(QUERY(temp <* HasProperties | SELF ::= temp)) = 0;

WR22 : IfcUniquePropertyName(HasProperties);

END_ENTITY;

ENTITY IfcSimpleProperty

ABSTRACT SUPERTYPE OF (ONEOF(IfcPropertyBoundedValue, IfcPropertyEnumeratedValue, IfcPropertyListValue, IfcPropertyReferenceValue, IfcPropertySingleValue, IfcPropertyTableValue))

SUBTYPE OF(IfcProperty);

END_ENTITY;

ENTITY IfcPropertyBoundedValue

SUBTYPE OF(IfcSimpleProperty);

UpperBoundValue : OPTIONAL IfcValue;

LowerBoundValue : OPTIONAL IfcValue;

Unit : OPTIONAL IfcUnit;

SetPointValue : OPTIONAL IfcValue;

WHERE

SAMEUNITUPPERLOWER : NOT(EXISTS(UpperBoundValue)) OR NOT(EXISTS(LowerBoundValue)) OR
(TYPEOF(UpperBoundValue) = TYPEOF(LowerBoundValue));

SAMEUNITUPPERSET : NOT(EXISTS(UpperBoundValue)) OR NOT(EXISTS(SetPointValue)) OR
(TYPEOF(UpperBoundValue) = TYPEOF(SetPointValue));

SAMEUNITLOWERSET : NOT(EXISTS(LowerBoundValue)) OR NOT(EXISTS(SetPointValue)) OR
(TYPEOF(LowerBoundValue) = TYPEOF(SetPointValue));

END_ENTITY;

ENTITY IfcPropertyEnumeratedValue

SUBTYPE OF(IfcSimpleProperty);

EnumerationValues : OPTIONAL LIST [1:?] OF IfcValue;

EnumerationReference : OPTIONAL IfcPropertyEnumeration;

WHERE

WR21 : NOT(EXISTS(EnumerationReference))
OR NOT(EXISTS(EnumerationValues))
OR (SIZEOF(QUERY(temp <* EnumerationValues |
temp IN EnumerationReference.EnumerationValues))
= SIZEOF(EnumerationValues));

END_ENTITY;

ENTITY IfcPropertyListValue

SUBTYPE OF(IfcSimpleProperty);

ListValues : OPTIONAL LIST [1:?] OF IfcValue;

Unit : OPTIONAL IfcUnit;

WHERE

WR31 : SIZEOF(QUERY(temp <* SELF.ListValues |

```
NOT(TYPEOF(SELF.ListValues[1]) = TYPEOF(temp))
)) = 0;
```

```
END_ENTITY;
```

```
ENTITY IfcPropertyReferenceValue
```

```
  SUBTYPE OF(IfcSimpleProperty);
```

```
  UsageName          : OPTIONAL IfcText;
```

```
  PropertyReference : OPTIONAL IfcObjectReferenceSelect;
```

```
END_ENTITY;
```

```
ENTITY IfcExternalReference
```

```
  ABSTRACT SUPERTYPE OF (ONEOF(IfcClassificationReference, IfcDocumentReference,
IfcExternallyDefinedHatchStyle, IfcExternallyDefinedSurfaceStyle, IfcExternallyDefinedTextFont,
IfcLibraryReference));
```

```
  Location              : OPTIONAL IfcURIReference;
```

```
  Identification        : OPTIONAL IfcIdentifier;
```

```
  Name                  : OPTIONAL IfcLabel;
```

```
  INVERSE
```

```
  ExternalReferenceForResources : SET OF IfcExternalReferenceRelationship FOR
RelatingReference;
```

```
  WHERE
```

```
  WR1 : EXISTS(Identification) OR EXISTS(Location) OR EXISTS(Name);
```

```
END_ENTITY;
```

```
ENTITY IfcClassificationReference
```

```
  SUBTYPE OF(IfcExternalReference);
```

```
  ReferencedSource      : OPTIONAL IfcClassificationReferenceSelect;
```

```
  Description           : OPTIONAL IfcText;
```

```
  Sort                  : OPTIONAL IfcIdentifier;
```

```
  INVERSE
```

```
  ClassificationRefForObjects : SET OF IfcRelAssociatesClassification FOR
```


RelatingClassification;

HasReferences : SET OF IfcClassificationReference FOR ReferencedSource;

END_ENTITY;

ENTITY IfcClassification

SUBTYPE OF(IfcExternalInformation);

Source : OPTIONAL IfcLabel;

Edition : OPTIONAL IfcLabel;

EditionDate : OPTIONAL IfcDate;

Name : IfcLabel;

Description : OPTIONAL IfcText;

Location : OPTIONAL IfcURIReference;

ReferenceTokens : OPTIONAL LIST [1:?] OF IfcIdentifier;

INVERSE

ClassificationForObjects : SET OF IfcRelAssociatesClassification FOR RelatingClassification;

HasReferences : SET OF IfcClassificationReference FOR ReferencedSource;

END_ENTITY;

ENTITY IfcExternalInformation

ABSTRACT SUPERTYPE OF (ONEOF(IfcClassification, IfcDocumentInformation, IfcLibraryInformation));

END_ENTITY;

ENTITY IfcDocumentInformation

SUBTYPE OF(IfcExternalInformation);

Identification : IfcIdentifier;

Name : IfcLabel;

Description : OPTIONAL IfcText;

Location : OPTIONAL IfcURIReference;

Purpose : OPTIONAL IfcText;

IntendedUse : OPTIONAL IfcText;
 Scope : OPTIONAL IfcText;
 Revision : OPTIONAL IfcLabel;
 DocumentOwner : OPTIONAL IfcActorSelect;
 Editors : OPTIONAL SET [1:?] OF IfcActorSelect;
 CreationTime : OPTIONAL IfcDateTime;
 LastRevisionTime : OPTIONAL IfcDateTime;
 ElectronicFormat : OPTIONAL IfcIdentifier;
 ValidFrom : OPTIONAL IfcDate;
 ValidUntil : OPTIONAL IfcDate;
 Confidentiality : OPTIONAL IfcDocumentConfidentialityEnum;
 Status : OPTIONAL IfcDocumentStatusEnum;

INVERSE

DocumentInfoForObjects : SET OF IfcRelAssociatesDocument FOR RelatingDocument;
 HasDocumentReferences : SET OF IfcDocumentReference FOR ReferencedDocument;
 IsPointedTo : SET OF IfcDocumentInformationRelationship FOR RelatedDocuments;
 IsPointer : SET [0:1] OF IfcDocumentInformationRelationship FOR

RelatingDocument;

END_ENTITY;

ENTITY IfcRelAssociatesDocument

SUBTYPE OF(IfcRelAssociates);

RelatingDocument : IfcDocumentSelect;

END_ENTITY;

ENTITY IfcDocumentReference

SUBTYPE OF(IfcExternalReference);

Description : OPTIONAL IfcText;

ReferencedDocument : OPTIONAL IfcDocumentInformation;

INVERSE

```
DocumentRefForObjects : SET OF IfcRelAssociatesDocument FOR RelatingDocument;  
WHERE  
WR1 : EXISTS(Name) XOR EXISTS(ReferencedDocument);  
END_ENTITY;
```

```
ENTITY IfcDocumentInformationRelationship  
SUBTYPE OF(IfcResourceLevelRelationship);  
RelatingDocument : IfcDocumentInformation;  
RelatedDocuments : SET [1:?] OF IfcDocumentInformation;  
RelationshipType : OPTIONAL IfcLabel;  
END_ENTITY;
```

```
ENTITY IfcLibraryInformation  
SUBTYPE OF(IfcExternalInformation);  
Name : IfcLabel;  
Version : OPTIONAL IfcLabel;  
Publisher : OPTIONAL IfcActorSelect;  
VersionDate : OPTIONAL IfcDateTime;  
Location : OPTIONAL IfcURIReference;  
Description : OPTIONAL IfcText;  
INVERSE  
LibraryInfoForObjects : SET OF IfcRelAssociatesLibrary FOR RelatingLibrary;  
HasLibraryReferences : SET OF IfcLibraryReference FOR ReferencedLibrary;  
END_ENTITY;
```

```
ENTITY IfcRelAssociatesLibrary  
SUBTYPE OF(IfcRelAssociates);  
RelatingLibrary : IfcLibrarySelect;  
END_ENTITY;
```

```

ENTITY IfcLibraryReference
  SUBTYPE OF(IfcExternalReference);
  Description      : OPTIONAL IfcText;
  Language         : OPTIONAL IfcLanguageId;
  ReferencedLibrary : OPTIONAL IfcLibraryInformation;
  INVERSE
  LibraryRefForObjects : SET OF IfcRelAssociatesLibrary FOR RelatingLibrary;
END_ENTITY;

```

```

ENTITY IfcRelAssociatesClassification
  SUBTYPE OF(IfcRelAssociates);
  RelatingClassification : IfcClassificationSelect;
END_ENTITY;

```

```

ENTITY IfcExternallyDefinedHatchStyle
  SUBTYPE OF(IfcExternalReference);
END_ENTITY;

```

```

ENTITY IfcExternallyDefinedSurfaceStyle
  SUBTYPE OF(IfcExternalReference);
END_ENTITY;

```

```

ENTITY IfcExternallyDefinedTextFont
  SUBTYPE OF(IfcExternalReference);
END_ENTITY;

```

```

ENTITY IfcTable:
  Name      : OPTIONAL IfcLabel;
  Rows      : OPTIONAL LIST [1:?] OF IfcTableRow;
  Columns   : OPTIONAL LIST [1:?] OF IfcTableColumn;

```

```

DERIVE
  NumberOfCellsInRow : IfcInteger := HIINDEX(Rows[1].RowCells);
  NumberOfHeadings   : IfcInteger := SIZEOF(QUERY( Temp <* Rows | Temp.IsHeading));
  NumberOfDataRows   : IfcInteger := SIZEOF(QUERY( Temp <* Rows | NOT(Temp.IsHeading)));
WHERE
  WR1 : SIZEOF(QUERY( Temp <* Rows | HIINDEX(Temp.RowCells) <> HIINDEX(Rows[1].RowCells))) =
0;
  WR2 : { 0 <= NumberOfHeadings <= 1 };
END_ENTITY;

ENTITY IfcTableRow;
  RowCells : OPTIONAL LIST [1:?] OF IfcValue;
  IsHeading : OPTIONAL IfcBoolean;
END_ENTITY;

ENTITY IfcTableColumn;
  Identifier : OPTIONAL IfcIdentifier;
  Name : OPTIONAL IfcLabel;
  Description : OPTIONAL IfcText;
  Unit : OPTIONAL IfcUnit;
  ReferencePath : OPTIONAL IfcReference;
END_ENTITY;

ENTITY IfcTimeSeries
  ABSTRACT SUPERTYPE OF (ONEOF(IfcIrregularTimeSeries, IfcRegularTimeSeries));
  Name : IfcLabel;
  Description : OPTIONAL IfcText;
  StartTime : IfcDateTime;
  EndTime : IfcDateTime;
  TimeSeriesDataType : IfcTimeSeriesDataTypeEnum;

```

```

    DataOrigin          : IfcDataOriginEnum;
    UserDefinedDataOrigin : OPTIONAL IfcLabel;
    Unit                : OPTIONAL IfcUnit;
    INVERSE
    HasExternalReference : SET [1:?] OF IfcExternalReferenceRelationship FOR
    RelatedResourceObjects;
    END_ENTITY;

```

```

ENTITY IfcIrregularTimeSeries
    SUBTYPE OF(IfcTimeSeries);
    Values : LIST [1:?] OF IfcIrregularTimeSeriesValue;
    END_ENTITY;

```

```

ENTITY IfcIrregularTimeSeriesValue;
    TimeStamp : IfcDateTime;
    ListValues : LIST [1:?] OF IfcValue;
    END_ENTITY;

```

```

ENTITY IfcRegularTimeSeries
    SUBTYPE OF(IfcTimeSeries);
    TimeStep : IfcTimeMeasure;
    Values : LIST [1:?] OF IfcTimeSeriesValue;
    END_ENTITY;

```

```

ENTITY IfcTimeSeriesValue;
    ListValues : LIST [1:?] OF IfcValue;
    END_ENTITY;

```

```

ENTITY IfcPropertySingleValue
    SUBTYPE OF(IfcSimpleProperty);

```

```

    NominalValue : OPTIONAL IfcValue;
    Unit          : OPTIONAL IfcUnit;
END_ENTITY;

```

```

ENTITY IfcPropertyTableValue

```

```

    SUBTYPE OF(IfcSimpleProperty);

```

```

    DefiningValues : OPTIONAL LIST [1:?] OF IfcValue;

```

```

    DefinedValues  : OPTIONAL LIST [1:?] OF IfcValue;

```

```

    Expression     : OPTIONAL IfcText;

```

```

    DefiningUnit   : OPTIONAL IfcUnit;

```

```

    DefinedUnit    : OPTIONAL IfcUnit;

```

```

    CurveInterpolation : OPTIONAL IfcCurveInterpolationEnum;

```

```

WHERE

```

```

    WR21 : (NOT(EXISTS(DefiningValues)) AND NOT(EXISTS(DefinedValues)))

```

```

        OR (SIZEOF(DefiningValues) = SIZEOF(DefinedValues));

```

```

    WR22 : NOT(EXISTS(DefiningValues)) OR

```

```

        (SIZEOF(QUERY(temp <*> SELF.DefiningValues | TYPEOF(temp) <>
TYPEOF(SELF.DefiningValues[1])
        )) = 0);

```

```

    WR23 : NOT(EXISTS(DefinedValues)) OR

```

```

        (SIZEOF(QUERY(temp <*> SELF.DefinedValues | TYPEOF(temp) <>
TYPEOF(SELF.DefinedValues[1])
        )) = 0);

```

```

END_ENTITY;

```

```

ENTITY IfcPropertySet

```

```

    SUBTYPE OF(IfcPropertySetDefinition);

```

```

    HasProperties : SET [1:?] OF IfcProperty;

```

```

WHERE

```

```

    EXISTSNAME : EXISTS(SELFWIfcRoot.Name);

```

```
    UNIQUEPROPERTYNAMES : IfcUniquePropertyName(HasProperties);  
END_ENTITY;
```

```
ENTITY IfcPropertyDependencyRelationship  
    SUBTYPE OF(IfcResourceLevelRelationship);  
    DependingProperty : IfcProperty;  
    DependantProperty : IfcProperty;  
    Expression        : OPTIONAL IfcText;  
  
    WHERE  
        NOSELFREFERENCE : DependingProperty <> DependantProperty;  
END_ENTITY;
```

```
ENTITY IfcConstraint  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcMetric, IfcObjective));  
    Name                : IfcLabel;  
    Description         : OPTIONAL IfcText;  
    ConstraintGrade     : IfcConstraintEnum;  
    ConstraintSource    : OPTIONAL IfcLabel;  
    CreatingActor       : OPTIONAL IfcActorSelect;  
    CreationTime        : OPTIONAL IfcDateTime;  
    UserDefinedGrade    : OPTIONAL IfcLabel;  
  
    INVERSE  
        HasExternalReferences : SET OF IfcExternalReferenceRelationship FOR RelatedResourceObjects;  
        PropertiesForConstraint : SET OF IfcResourceConstraintRelationship FOR RelatingConstraint;  
  
    WHERE  
        WR11 : (ConstraintGrade <> IfcConstraintEnum.USERDEFINED) OR  
                ((ConstraintGrade = IfcConstraintEnum.USERDEFINED) AND  
                EXISTS(SELFWIfcConstraint.UserDefinedGrade));  
END_ENTITY;
```


ENTITY IfcMetric

SUBTYPE OF(IfcConstraint);

Benchmark : IfcBenchmarkEnum;
ValueSource : OPTIONAL IfcLabel;
DataValue : OPTIONAL IfcMetricValueSelect;
ReferencePath : OPTIONAL IfcReference;

END_ENTITY;

ENTITY IfcObjective

SUBTYPE OF(IfcConstraint);

BenchmarkValues : OPTIONAL LIST [1:?] OF IfcConstraint;
LogicalAggregator : OPTIONAL IfcLogicalOperatorEnum;
ObjectiveQualifier : IfcObjectiveEnum;
UserDefinedQualifier : OPTIONAL IfcLabel;

WHERE

WR21 : (ObjectiveQualifier <> IfcObjectiveEnum.USERDEFINED) OR

((ObjectiveQualifier = IfcObjectiveEnum.USERDEFINED) AND

EXISTS(SELFWIfcObjective.UserDefinedQualifier));

END_ENTITY;

ENTITY IfcResourceConstraintRelationship

SUBTYPE OF(IfcResourceLevelRelationship);

RelatingConstraint : IfcConstraint;
RelatedResourceObjects : SET [1:?] OF IfcResourceObjectSelect;

END_ENTITY;

ENTITY IfcResourceApprovalRelationship

SUBTYPE OF(IfcResourceLevelRelationship);

RelatedResourceObjects : SET [1:?] OF IfcResourceObjectSelect;
RelatingApproval : IfcApproval;

END_ENTITY;

ENTITY IfcPreDefinedProperties

ABSTRACT SUPERTYPE OF (ONEOF(IfcReinforcementBarProperties, IfcSectionProperties,
IfcSectionReinforcementProperties))

SUBTYPE OF(IfcPropertyAbstraction);

END_ENTITY;

ENTITY IfcReinforcementBarProperties

SUBTYPE OF(IfcPreDefinedProperties);

TotalCrossSectionArea : IfcAreaMeasure;

SteelGrade : IfcLabel;

BarSurface : OPTIONAL IfcReinforcingBarSurfaceEnum;

EffectiveDepth : OPTIONAL IfcLengthMeasure;

NominalBarDiameter : OPTIONAL IfcPositiveLengthMeasure;

BarCount : OPTIONAL IfcCountMeasure;

END_ENTITY;

ENTITY IfcSectionProperties

SUBTYPE OF(IfcPreDefinedProperties);

SectionType : IfcSectionTypeEnum;

StartProfile : IfcProfileDef;

EndProfile : OPTIONAL IfcProfileDef;

END_ENTITY;

ENTITY IfcSectionReinforcementProperties

SUBTYPE OF(IfcPreDefinedProperties);

LongitudinalStartPosition : IfcLengthMeasure;

LongitudinalEndPosition : IfcLengthMeasure;

TransversePosition : OPTIONAL IfcLengthMeasure;

```

ReinforcementRole          : IfcReinforcingBarRoleEnum;
SectionDefinition          : IfcSectionProperties;
CrossSectionReinforcementDefinitions : SET [1:?] OF IfcReinforcementBarProperties;
END_ENTITY;

```

```

ENTITY IfcDoorStyle
  SUBTYPE OF(IfcTypeProduct);
  OperationType            : IfcDoorStyleOperationEnum;
  ConstructionType        : IfcDoorStyleConstructionEnum;
  ParameterTakesPrecedence : IfcBoolean;
  Sizeable                 : IfcBoolean;
END_ENTITY;

```

```

ENTITY IfcElementType
  ABSTRACT SUPERTYPE OF (ONEOF(IfcBuildingElementType, IfcCivilElementType,
IfcDistributionElementType, IfcElementAssemblyType, IfcElementComponentType,
IfcFurnishingElementType, IfcGeographicElementType, IfcTransportElementType))
  SUBTYPE OF(IfcTypeProduct);
  ElementType : OPTIONAL IfcLabel;
END_ENTITY;

```

```

ENTITY IfcBuildingElementType
  ABSTRACT SUPERTYPE OF (ONEOF(IfcBeamType, IfcBuildingElementProxyType, IfcChimneyType,
IfcColumnType, IfcCoveringType, IfcCurtainWallType, IfcDoorType, IfcFootingType, IfcMemberType,
IfcPileType, IfcPlateType, IfcRailingType, IfcRampFlightType, IfcRampType, IfcRoofType,
IfcShadingDeviceType, IfcSlabType, IfcStairFlightType, IfcStairType, IfcWallType, IfcWindowType))
  SUBTYPE OF(IfcElementType);
END_ENTITY;

```

```

ENTITY IfcBeamType
  SUBTYPE OF(IfcBuildingElementType);

```

```

    PredefinedType : IfcBeamTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcBeamTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcBeamTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcBuildingElementProxyType
    SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcBuildingElementProxyTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcBuildingElementProxyTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcBuildingElementProxyTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcChimneyType
    SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcChimneyTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcChimneyTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcChimneyTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcColumnType
    SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcColumnTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcColumnTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcColumnTypeEnum.USERDEFINED) AND

```

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcCoveringType

SUBTYPE OF(IfcBuildingElementType);

PredefinedType : IfcCoveringTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCoveringTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCoveringTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcCurtainWallType

SUBTYPE OF(IfcBuildingElementType);

PredefinedType : IfcCurtainWallTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCurtainWallTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCurtainWallTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcDoorType

SUBTYPE OF(IfcBuildingElementType);

PredefinedType : IfcDoorTypeEnum;

OperationType : IfcDoorTypeOperationEnum;

ParameterTakesPrecedence : OPTIONAL IfcBoolean;

UserDefinedOperationType : OPTIONAL IfcLabel;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDoorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDoorTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcFootingType

SUBTYPE OF(IfcBuildingElementType);

PredefinedType : IfcFootingTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFootingTypeEnum.USERDEFINED) OR

((PredefinedType = IfcFootingTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcMemberType

SUBTYPE OF(IfcBuildingElementType);

PredefinedType : IfcMemberTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcMemberTypeEnum.USERDEFINED) OR

((PredefinedType = IfcMemberTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcPileType

SUBTYPE OF(IfcBuildingElementType);

PredefinedType : IfcPileTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcPileTypeEnum.USERDEFINED) OR

((PredefinedType = IfcPileTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcPlateType

SUBTYPE OF(IfcBuildingElementType);

```

    PredefinedType : IfcPlateTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcPlateTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcPlateTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcRailingType
    SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcRailingTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcRailingTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcRailingTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcRampFlightType
    SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcRampFlightTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcRampFlightTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcRampFlightTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcRampType
    SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcRampTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcRampTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcRampTypeEnum.USERDEFINED) AND

```

```
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;
```

```
ENTITY IfcRoofType
  SUBTYPE OF(IfcBuildingElementType);
  PredefinedType : IfcRoofTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcRoofTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcRoofTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;
```

```
ENTITY IfcShadingDeviceType
  SUBTYPE OF(IfcBuildingElementType);
  PredefinedType : IfcShadingDeviceTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcShadingDeviceTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcShadingDeviceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;
```

```
ENTITY IfcSlabType
  SUBTYPE OF(IfcBuildingElementType);
  PredefinedType : IfcSlabTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSlabTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcSlabTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;
```

```
ENTITY IfcStairFlightType
```



```

SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcStairFlightTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcStairFlightTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcStairFlightTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcBuildingElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcStairType
SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcStairTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcStairTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcStairTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcBuildingElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcWallType
SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcWallTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcWallTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcWallTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcBuildingElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcWindowType
SUBTYPE OF(IfcBuildingElementType);
    PredefinedType : IfcWindowTypeEnum;
    PartitioningType : IfcWindowTypePartitioningEnum;
    ParameterTakesPrecedence : OPTIONAL IfcBoolean;

```

```

UserDefinedPartitioningType : OPTIONAL IfcLabel;
WHERE
CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcWindowTypeEnum.USERDEFINED) OR
((PredefinedType = IfcWindowTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcCivilElementType
ABSTRACT SUPERTYPE OF (ONEOF(IfcSubsidiaryFacilityType_K, IfcEarthworkElementType_K,
IfcRoadElementType_K, IfcCivilStructureElementType_K, IfcCivilElementProxyType_K))
SUBTYPE OF(IfcElementType);
END_ENTITY;

```

```

ENTITY IfcSubsidiaryFacilityType_K
ABSTRACT SUPERTYPE OF (ONEOF(IfcRoadSignEquipmentType_K, IfcPavementAdditionType_K,
IfcGuardType_K))
SUBTYPE OF(IfcCivilElementType);
END_ENTITY;

```

```

ENTITY IfcRoadSignEquipmentType_K
SUBTYPE OF(IfcSubsidiaryFacilityType_K);
PredefinedType : IfcRoadSignEquipmentTypeEnum_K;
END_ENTITY;

```

```

ENTITY IfcPavementAdditionType_K
SUBTYPE OF(IfcSubsidiaryFacilityType_K);
PredefinedType : IfcPavementAdditionTypeEnum_K;
END_ENTITY;

```

```

ENTITY IfcGuardType_K
SUBTYPE OF(IfcSubsidiaryFacilityType_K);

```

```
    PredefinedType : IfcGuardTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcEarthworkElementType_K  
    SUBTYPE OF(IfcCivilElementType);  
    PredefinedType : IfcEarthworkElementTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcRoadElementType_K  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcRoadShoulderType_K, IfcRoadBodyType_K,  
IfcRoadMedianStripType_K, IfcCurbType_K, IfcRoadPavementType_K))  
    SUBTYPE OF(IfcCivilElementType);  
END_ENTITY;
```

```
ENTITY IfcRoadShoulderType_K  
    SUBTYPE OF(IfcRoadElementType_K);  
    PredefinedType : IfcRoadShoulderTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcRoadBodyType_K  
    SUBTYPE OF(IfcRoadElementType_K);  
    PredefinedType : IfcRoadBodyTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcRoadMedianStripType_K  
    SUBTYPE OF(IfcRoadElementType_K);  
    PredefinedType : IfcRoadMedianStripTypeEnum;  
END_ENTITY;
```

```
ENTITY IfcCurbType_K
```

```
SUBTYPE OF(IfcRoadElementType_K);  
    PredefinedType : IfcCurbTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcRoadPavementType_K  
    SUBTYPE OF(IfcRoadElementType_K);  
    PredefinedType : IfcPavementTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcCivilStructureElementType_K  
    ABSTRACT SUPERTYPE OF (ONEOF(IfcRetainingWallType_K, IfcCulvertType_K, IfcCaissonType_K,  
IfcBridgeElementType_K, IfcTunnelElementType_K))  
    SUBTYPE OF(IfcCivilElementType);  
END_ENTITY;
```

```
ENTITY IfcRetainingWallType_K  
    SUBTYPE OF(IfcCivilStructureElementType_K);  
    PredefinedType : IfcRetainingWallTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcCulvertType_K  
    SUBTYPE OF(IfcCivilStructureElementType_K);  
    PredefinedType : IfcCulvertTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcCaissonType_K  
    SUBTYPE OF(IfcCivilStructureElementType_K);  
    PredefinedType : IfcCaissonTypeEnum_K;  
END_ENTITY;
```

```
ENTITY IfcBridgeElementType_K
    SUPERTYPE OF (ONEOF(IfcBridgeDeckType_K, IfcBridgeTowerType_K, IfcBridgeCableType_K,
IfcBridgePierType_K, IfcBridgeAbutmentType_K, IfcBridgeSpanType_K, IfcBridgeSegmentType_K,
IfcBridgeCopingType_K, IfcBridgeGirderType_K))
    SUBTYPE OF(IfcCivilStructureElementType_K);
END_ENTITY;
```

```
ENTITY IfcBridgeDeckType_K
    SUBTYPE OF(IfcBridgeElementType_K);
    PredefinedType : IfcBridgeDeckTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcBridgeTowerType_K
    SUBTYPE OF(IfcBridgeElementType_K);
    PredefinedType : IfcBridgeTowerTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcBridgeCableType_K
    SUBTYPE OF(IfcBridgeElementType_K);
    PredefinedType : IfcBridgeCableTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcBridgePierType_K
    SUBTYPE OF(IfcBridgeElementType_K);
    PredefinedType : IfcBridgePierTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcBridgeAbutmentType_K
    SUBTYPE OF(IfcBridgeElementType_K);
    PredefinedType : IfcBridgeAbutmentTypeEnum_K;
```

END_ENTITY;

ENTITY IfcBridgeSpanType_K

SUBTYPE OF(IfcBridgeElementType_K);

PredefinedType : IfcBridgeSpanTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeSegmentType_K

SUBTYPE OF(IfcBridgeElementType_K);

PredefinedType : IfcBridgeSegmentTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeCopingType_K

SUBTYPE OF(IfcBridgeElementType_K);

PredefinedType : IfcBridgeCopingTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeGirderType_K

SUBTYPE OF(IfcBridgeElementType_K);

PredefinedType : IfcBridgeGirderTypeEnum_K;

END_ENTITY;

ENTITY IfcTunnelElementType_K

ABSTRACT SUPERTYPE OF (ONEOF(IfcTunnelLiningType_K, IfcTunnelLiningSegmentType_K))

SUBTYPE OF(IfcCivilStructureElementType_K);

END_ENTITY;

ENTITY IfcTunnelLiningType_K

SUBTYPE OF(IfcTunnelElementType_K);

PredefinedType : IfcTunnelLiningTypeEnum_K;

END_ENTITY;

ENTITY IfcTunnelLiningSegmentType_K

SUBTYPE OF(IfcTunnelElementType_K);

PredefinedType : IfcTunnelLiningSegmentTypeEnum_K;

END_ENTITY;

ENTITY IfcCivilElementProxyType_K

SUBTYPE OF(IfcCivilElementType);

PredefinedType : IfcCivilElementProxyTypeEnum_K;

END_ENTITY;

ENTITY IfcDistributionElementType

SUPERTYPE OF (ONEOF(IfcDistributionControlElementType, IfcDistributionFlowElementType))

SUBTYPE OF(IfcElementType);

END_ENTITY;

ENTITY IfcDistributionControlElementType

ABSTRACT SUPERTYPE OF (ONEOF(IfcActuatorType, IfcAlarmType, IfcControllerType,
IfcFlowInstrumentType, IfcProtectiveDeviceTrippingUnitType, IfcSensorType,
IfcUnitaryControlElementType))

SUBTYPE OF(IfcDistributionElementType);

END_ENTITY;

ENTITY IfcActuatorType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcActuatorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcActuatorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcActuatorTypeEnum.USERDEFINED) AND

EXISTS(SELFIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcAlarmType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcAlarmTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcAlarmTypeEnum.USERDEFINED) OR

((PredefinedType = IfcAlarmTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcControllerType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcControllerTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcControllerTypeEnum.USERDEFINED) OR

((PredefinedType = IfcControllerTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcFlowInstrumentType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcFlowInstrumentTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFlowInstrumentTypeEnum.USERDEFINED) OR

((PredefinedType = IfcFlowInstrumentTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcProtectiveDeviceTrippingUnitType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcProtectiveDeviceTrippingUnitTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcProtectiveDeviceTrippingUnitTypeEnum.USERDEFINED) OR
((PredefinedType = IfcProtectiveDeviceTrippingUnitTypeEnum.USERDEFINED) AND EXISTS(SELFIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcSensorType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcSensorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSensorTypeEnum.USERDEFINED) OR
((PredefinedType = IfcSensorTypeEnum.USERDEFINED) AND EXISTS(SELFIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcUnitaryControlElementType

SUBTYPE OF(IfcDistributionControlElementType);

PredefinedType : IfcUnitaryControlElementTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcUnitaryControlElementTypeEnum.USERDEFINED) OR
((PredefinedType = IfcUnitaryControlElementTypeEnum.USERDEFINED) AND EXISTS(SELFIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcDistributionFlowElementType

ABSTRACT SUPERTYPE OF (ONEOF(IfcDistributionChamberElementType, IfcEnergyConversionDeviceType, IfcFlowControllerType, IfcFlowFittingType, IfcFlowMovingDeviceType, IfcFlowSegmentType, IfcFlowStorageDeviceType, IfcFlowTerminalType, IfcFlowTreatmentDeviceType))

SUBTYPE OF(IfcDistributionElementType);

END_ENTITY;

ENTITY IfcDistributionChamberElementType

SUBTYPE OF(IfcDistributionFlowElementType);

PredefinedType : IfcDistributionChamberElementTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDistributionChamberElementTypeEnum.USERDEFINED) OR
((PredefinedType = IfcDistributionChamberElementTypeEnum.USERDEFINED) AND EXISTS(SELFIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcEnergyConversionDeviceType

ABSTRACT SUPERTYPE OF (ONEOF(IfcAirToAirHeatRecoveryType, IfcBoilerType, IfcBurnerType, IfcChillerType, IfcCoilType, IfcCondenserType, IfcCooledBeamType, IfcCoolingTowerType, IfcElectricGeneratorType, IfcElectricMotorType, IfcEngineType, IfcEvaporativeCoolerType, IfcEvaporatorType, IfcHeatExchangerType, IfcHumidifierType, IfcMotorConnectionType, IfcSolarDeviceType, IfcTransformerType, IfcTubeBundleType, IfcUnitaryEquipmentType))

SUBTYPE OF(IfcDistributionFlowElementType);

END_ENTITY;

ENTITY IfcAirToAirHeatRecoveryType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcAirToAirHeatRecoveryTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) OR
((PredefinedType = IfcAirToAirHeatRecoveryTypeEnum.USERDEFINED) AND EXISTS(SELFIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcBoilerType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcBoilerTypeEnum;

```

WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcBoilerTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcBoilerTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcBurnerType

```

```

    SUBTYPE OF(IfcEnergyConversionDeviceType);

```

```

    PredefinedType : IfcBurnerTypeEnum;

```

```

WHERE

```

```

    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcBurnerTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcBurnerTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcChillerType

```

```

    SUBTYPE OF(IfcEnergyConversionDeviceType);

```

```

    PredefinedType : IfcChillerTypeEnum;

```

```

WHERE

```

```

    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcChillerTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcChillerTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcCoilType

```

```

    SUBTYPE OF(IfcEnergyConversionDeviceType);

```

```

    PredefinedType : IfcCoilTypeEnum;

```

```

WHERE

```

```

    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCoilTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcCoilTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

END_ENTITY;

ENTITY IfcCondenserType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcCondenserTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCondenserTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCondenserTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcCooledBeamType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcCooledBeamTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCooledBeamTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCooledBeamTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcCoolingTowerType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcCoolingTowerTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCoolingTowerTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCoolingTowerTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcElectricGeneratorType

SUBTYPE OF(IfcEnergyConversionDeviceType);

```

    PredefinedType : IfcElectricGeneratorTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElectricGeneratorTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcElectricGeneratorTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcElectricMotorType
    SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcElectricMotorTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElectricMotorTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcElectricMotorTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcEngineType
    SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcEngineTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcEngineTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcEngineTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcEvaporativeCoolerType
    SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcEvaporativeCoolerTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcEvaporativeCoolerTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcEvaporativeCoolerTypeEnum.USERDEFINED) AND

```

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcEvaporatorType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcEvaporatorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcEvaporatorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcEvaporatorTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcHeatExchangerType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcHeatExchangerTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcHeatExchangerTypeEnum.USERDEFINED) OR

((PredefinedType = IfcHeatExchangerTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcHumidifierType

SUBTYPE OF(IfcEnergyConversionDeviceType);

PredefinedType : IfcHumidifierTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcHumidifierTypeEnum.USERDEFINED) OR

((PredefinedType = IfcHumidifierTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcMotorConnectionType

```

SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcMotorConnectionTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcMotorConnectionTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcMotorConnectionTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcSolarDeviceType
    SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcSolarDeviceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSolarDeviceTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcSolarDeviceTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcTransformerType
    SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcTransformerTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTransformerTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcTransformerTypeEnum.USERDEFINED) AND
EXISTS(SELFIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcTubeBundleType
    SUBTYPE OF(IfcEnergyConversionDeviceType);
    PredefinedType : IfcTubeBundleTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTubeBundleTypeEnum.USERDEFINED) OR

```

```
                ((PredefinedType = IfcTubeBundleTypeEnum.USERDEFINED) AND  
EXISTS(SELFWIfcElementType.ElementType));
```

```
END_ENTITY;
```

```
ENTITY IfcUnitaryEquipmentType
```

```
    SUBTYPE OF(IfcEnergyConversionDeviceType);
```

```
    PredefinedType : IfcUnitaryEquipmentTypeEnum;
```

```
WHERE
```

```
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcUnitaryEquipmentTypeEnum.USERDEFINED) OR
```

```
                ((PredefinedType = IfcUnitaryEquipmentTypeEnum.USERDEFINED) AND  
EXISTS(SELFWIfcElementType.ElementType));
```

```
END_ENTITY;
```

```
ENTITY IfcFlowControllerType
```

```
    ABSTRACT SUPERTYPE OF (ONEOF(IfcAirTerminalBoxType, IfcDamperType,  
IfcElectricDistributionBoardType, IfcElectricTimeControlType, IfcFlowMeterType,  
IfcProtectiveDeviceType, IfcSwitchingDeviceType, IfcValveType))
```

```
    SUBTYPE OF(IfcDistributionFlowElementType);
```

```
END_ENTITY;
```

```
ENTITY IfcAirTerminalBoxType
```

```
    SUBTYPE OF(IfcFlowControllerType);
```

```
    PredefinedType : IfcAirTerminalBoxTypeEnum;
```

```
WHERE
```

```
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcAirTerminalBoxTypeEnum.USERDEFINED) OR
```

```
                ((PredefinedType = IfcAirTerminalBoxTypeEnum.USERDEFINED) AND  
EXISTS(SELFWIfcElementType.ElementType));
```

```
END_ENTITY;
```

```
ENTITY IfcDamperType
```

```
    SUBTYPE OF(IfcFlowControllerType);
```



```

    PredefinedType : IfcDamperTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDamperTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcDamperTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcElectricDistributionBoardType
    SUBTYPE OF(IfcFlowControllerType);
    PredefinedType : IfcElectricDistributionBoardTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElectricDistributionBoardTypeEnum.USERDEFINED)
OR
                            ((PredefinedType = IfcElectricDistributionBoardTypeEnum.USERDEFINED)
AND EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcElectricTimeControlType
    SUBTYPE OF(IfcFlowControllerType);
    PredefinedType : IfcElectricTimeControlTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElectricTimeControlTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcElectricTimeControlTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcFlowMeterType
    SUBTYPE OF(IfcFlowControllerType);
    PredefinedType : IfcFlowMeterTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFlowMeterTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcFlowMeterTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcProtectiveDeviceType
  SUBTYPE OF(IfcFlowControllerType);
  PredefinedType : IfcProtectiveDeviceTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcProtectiveDeviceTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcProtectiveDeviceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcSwitchingDeviceType
  SUBTYPE OF(IfcFlowControllerType);
  PredefinedType : IfcSwitchingDeviceTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSwitchingDeviceTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcSwitchingDeviceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcValveType
  SUBTYPE OF(IfcFlowControllerType);
  PredefinedType : IfcValveTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcValveTypeEnum.USERDEFINED) OR
                ((PredefinedType = IfcValveTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcFlowFittingType
  ABSTRACT SUPERTYPE OF (ONEOF(IfcCableCarrierFittingType, IfcCableFittingType,
IfcDuctFittingType, IfcJunctionBoxType, IfcPipeFittingType, IfcGutterFittingType_K))
  SUBTYPE OF(IfcDistributionFlowElementType);
END_ENTITY;

```

```

ENTITY IfcCableCarrierFittingType
  SUBTYPE OF(IfcFlowFittingType);
  PredefinedType : IfcCableCarrierFittingTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCableCarrierFittingTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcCableCarrierFittingTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcCableFittingType
  SUBTYPE OF(IfcFlowFittingType);
  PredefinedType : IfcCableFittingTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCableFittingTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcCableFittingTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcDuctFittingType
  SUBTYPE OF(IfcFlowFittingType);
  PredefinedType : IfcDuctFittingTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDuctFittingTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcDuctFittingTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

END_ENTITY;

ENTITY IfcJunctionBoxType

SUBTYPE OF(IfcFlowFittingType);

PredefinedType : IfcJunctionBoxTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcJunctionBoxTypeEnum.USERDEFINED) OR

((PredefinedType = IfcJunctionBoxTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcPipeFittingType

SUBTYPE OF(IfcFlowFittingType);

PredefinedType : IfcPipeFittingTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcPipeFittingTypeEnum.USERDEFINED) OR

((PredefinedType = IfcPipeFittingTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcGutterFittingType_K

SUBTYPE OF(IfcFlowFittingType);

PredefinedType : IfcGutterFittingTypeEnum_K;

END_ENTITY;

ENTITY IfcFlowMovingDeviceType

ABSTRACT SUPERTYPE OF (ONEOF(IfcCompressorType, IfcFanType, IfcPumpType))

SUBTYPE OF(IfcDistributionFlowElementType);

END_ENTITY;

```

ENTITY IfcCompressorType
  SUBTYPE OF(IfcFlowMovingDeviceType);
  PredefinedType : IfcCompressorTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCompressorTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcCompressorTypeEnum.USERDEFINED) AND
    EXISTS(SELFWIfcElementType.ElementType));
  END_ENTITY;

```

```

ENTITY IfcFanType
  SUBTYPE OF(IfcFlowMovingDeviceType);
  PredefinedType : IfcFanTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFanTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcFanTypeEnum.USERDEFINED) AND
    EXISTS(SELFWIfcElementType.ElementType));
  END_ENTITY;

```

```

ENTITY IfcPumpType
  SUBTYPE OF(IfcFlowMovingDeviceType);
  PredefinedType : IfcPumpTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcPumpTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcPumpTypeEnum.USERDEFINED) AND
    EXISTS(SELFWIfcElementType.ElementType));
  END_ENTITY;

```

```

ENTITY IfcFlowSegmentType
  ABSTRACT SUPERTYPE OF (ONEOF(IfcCableCarrierSegmentType, IfcCableSegmentType,
IfcDuctSegmentType, IfcPipeSegmentType, IfcGutterSegmentType_K))
  SUBTYPE OF(IfcDistributionFlowElementType);

```

END_ENTITY;

ENTITY IfcCableCarrierSegmentType

SUBTYPE OF(IfcFlowSegmentType);

PredefinedType : IfcCableCarrierSegmentTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCableCarrierSegmentTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCableCarrierSegmentTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcCableSegmentType

SUBTYPE OF(IfcFlowSegmentType);

PredefinedType : IfcCableSegmentTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCableSegmentTypeEnum.USERDEFINED) OR

((PredefinedType = IfcCableSegmentTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcDuctSegmentType

SUBTYPE OF(IfcFlowSegmentType);

PredefinedType : IfcDuctSegmentTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDuctSegmentTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDuctSegmentTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcPipeSegmentType

SUBTYPE OF(IfcFlowSegmentType);

```

    PredefinedType : IfcPipeSegmentTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcPipeSegmentTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcPipeSegmentTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcGutterSegmentType_K
    SUBTYPE OF(IfcFlowSegmentType);
    PredefinedType : IfcGutterSegmentTypeEnum_K;
END_ENTITY;

```

```

ENTITY IfcFlowStorageDeviceType
    ABSTRACT SUPERTYPE OF (ONEOF(IfcElectricFlowStorageDeviceType, IfcTankType))
    SUBTYPE OF(IfcDistributionFlowElementType);
END_ENTITY;

```

```

ENTITY IfcElectricFlowStorageDeviceType
    SUBTYPE OF(IfcFlowStorageDeviceType);
    PredefinedType : IfcElectricFlowStorageDeviceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED)
OR
        ((PredefinedType = IfcElectricFlowStorageDeviceTypeEnum.USERDEFINED)
AND EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcTankType
    SUBTYPE OF(IfcFlowStorageDeviceType);
    PredefinedType : IfcTankTypeEnum;
WHERE

```

```

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTankTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcTankTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

```

END_ENTITY;

```

```

ENTITY IfcFlowTerminalType

```

```

  ABSTRACT SUPERTYPE OF (ONEOF(IfcAirTerminalType, IfcAudioVisualApplianceType,
IfcCommunicationsApplianceType, IfcElectricApplianceType, IfcFireSuppressionTerminalType,
IfcLampType, IfcLightFixtureType, IfcMedicalDeviceType, IfcOutletType, IfcSanitaryTerminalType,
IfcSpaceHeaterType, IfcStackTerminalType, IfcWasteTerminalType))

```

```

  SUBTYPE OF(IfcDistributionFlowElementType);

```

```

END_ENTITY;

```

```

ENTITY IfcAirTerminalType

```

```

  SUBTYPE OF(IfcFlowTerminalType);

```

```

  PredefinedType : IfcAirTerminalTypeEnum;

```

```

  WHERE

```

```

    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcAirTerminalTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcAirTerminalTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

```

END_ENTITY;

```

```

ENTITY IfcAudioVisualApplianceType

```

```

  SUBTYPE OF(IfcFlowTerminalType);

```

```

  PredefinedType : IfcAudioVisualApplianceTypeEnum;

```

```

  WHERE

```

```

    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcAudioVisualApplianceTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcAudioVisualApplianceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

```

END_ENTITY;

```

```

ENTITY IfcCommunicationsApplianceType

```



```

SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcCommunicationsApplianceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcCommunicationsApplianceTypeEnum.USERDEFINED)
OR
    ((PredefinedType = IfcCommunicationsApplianceTypeEnum.USERDEFINED)
AND EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcElectricApplianceType
    SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcElectricApplianceTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElectricApplianceTypeEnum.USERDEFINED) OR
    ((PredefinedType = IfcElectricApplianceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcFireSuppressionTerminalType
    SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcFireSuppressionTerminalTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFireSuppressionTerminalTypeEnum.USERDEFINED)
OR
    ((PredefinedType = IfcFireSuppressionTerminalTypeEnum.USERDEFINED)
AND EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcLampType
    SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcLampTypeEnum;

```

```

WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcLampTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcLampTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcLightFixtureType
    SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcLightFixtureTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcLightFixtureTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcLightFixtureTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcMedicalDeviceType
    SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcMedicalDeviceTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcMedicalDeviceTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcMedicalDeviceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcOutletType
    SUBTYPE OF(IfcFlowTerminalType);
    PredefinedType : IfcOutletTypeEnum;
    WHERE
        CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcOutletTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcOutletTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

END_ENTITY;

ENTITY IfcSanitaryTerminalType

SUBTYPE OF(IfcFlowTerminalType);

PredefinedType : IfcSanitaryTerminalTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSanitaryTerminalTypeEnum.USERDEFINED) OR

((PredefinedType = IfcSanitaryTerminalTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcSpaceHeaterType

SUBTYPE OF(IfcFlowTerminalType);

PredefinedType : IfcSpaceHeaterTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSpaceHeaterTypeEnum.USERDEFINED) OR

((PredefinedType = IfcSpaceHeaterTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcStackTerminalType

SUBTYPE OF(IfcFlowTerminalType);

PredefinedType : IfcStackTerminalTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcStackTerminalTypeEnum.USERDEFINED) OR

((PredefinedType = IfcStackTerminalTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcWasteTerminalType

SUBTYPE OF(IfcFlowTerminalType);

```

    PredefinedType : IfcWasteTerminalTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcWasteTerminalTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcWasteTerminalTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcFlowTreatmentDeviceType
    ABSTRACT SUPERTYPE OF (ONEOF(IfcDuctSilencerType, IfcFilterType, IfcInterceptorType))
    SUBTYPE OF(IfcDistributionFlowElementType);
END_ENTITY;

```

```

ENTITY IfcDuctSilencerType
    SUBTYPE OF(IfcFlowTreatmentDeviceType);
    PredefinedType : IfcDuctSilencerTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDuctSilencerTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcDuctSilencerTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcFilterType
    SUBTYPE OF(IfcFlowTreatmentDeviceType);
    PredefinedType : IfcFilterTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFilterTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcFilterTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcInterceptorType

```

```

SUBTYPE OF(IfcFlowTreatmentDeviceType);
    PredefinedType : IfcInterceptorTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcInterceptorTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcInterceptorTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcElementAssemblyType
    SUBTYPE OF(IfcElementType);
    PredefinedType : IfcElementAssemblyTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcElementAssemblyTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcElementAssemblyTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcElementComponentType
    ABSTRACT SUPERTYPE OF (ONEOF(IfcBuildingElementPartType, IfcDiscreteAccessoryType,
IfcFastenerType, IfcMechanicalFastenerType, IfcReinforcingElementType, IfcVibrationIsolatorType,
IfcWaterProofingElementType_K, IfcRoadElementPartType_K, IfcBridgeElementPartType_K,
IfcTunnelElementPartType_K))
    SUBTYPE OF(IfcElementType);
END_ENTITY;

```

```

ENTITY IfcBuildingElementPartType
    SUBTYPE OF(IfcElementComponentType);
    PredefinedType : IfcBuildingElementPartTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcBuildingElementPartTypeEnum.USERDEFINED) OR
        ((PredefinedType = IfcBuildingElementPartTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcElementType.ElementType));

```

END_ENTITY;

ENTITY IfcDiscreteAccessoryType

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcDiscreteAccessoryTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcDiscreteAccessoryTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDiscreteAccessoryTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcFastenerType

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcFastenerTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFastenerTypeEnum.USERDEFINED) OR

((PredefinedType = IfcFastenerTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcMechanicalFastenerType

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcMechanicalFastenerTypeEnum;

NominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

NominalLength : OPTIONAL IfcPositiveLengthMeasure;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcMechanicalFastenerTypeEnum.USERDEFINED) OR

((PredefinedType = IfcMechanicalFastenerTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

```

ENTITY IfcReinforcingElementType
  ABSTRACT SUPERTYPE OF (ONEOF(IfcReinforcingBarType, IfcReinforcingMeshType,
IfcTendonAnchorType, IfcTendonType) ANDOR IfcGroundReinforcingElementType_K)
  SUBTYPE OF(IfcElementComponentType);
END_ENTITY;

```

```

ENTITY IfcReinforcingBarType

```

```

  SUBTYPE OF(IfcReinforcingElementType);

```

```

  PredefinedType : IfcReinforcingBarTypeEnum;

```

```

  NominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

```

```

  CrossSectionArea : OPTIONAL IfcAreaMeasure;

```

```

  BarLength : OPTIONAL IfcPositiveLengthMeasure;

```

```

  BarSurface : OPTIONAL IfcReinforcingBarSurfaceEnum;

```

```

  BendingShapeCode : OPTIONAL IfcLabel;

```

```

  BendingParameters : OPTIONAL LIST [1:?] OF IfcBendingParameterSelect;

```

```

WHERE

```

```

  CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcReinforcingBarTypeEnum.USERDEFINED) OR

```

```

    ((PredefinedType = IfcReinforcingBarTypeEnum.USERDEFINED) AND

```

```

    EXISTS(SELFIfcElementType.ElementType));

```

```

  BENDINGSHAPECODEPROVIDED : NOT EXISTS(BendingParameters) OR EXISTS(BendingShapeCode);

```

```

END_ENTITY;

```

```

ENTITY IfcReinforcingMeshType

```

```

  SUBTYPE OF(IfcReinforcingElementType);

```

```

  PredefinedType : IfcReinforcingMeshTypeEnum;

```

```

  MeshLength : OPTIONAL IfcPositiveLengthMeasure;

```

```

  MeshWidth : OPTIONAL IfcPositiveLengthMeasure;

```

```

  LongitudinalBarNominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

```

```

  TransverseBarNominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

```

```

  LongitudinalBarCrossSectionArea : OPTIONAL IfcAreaMeasure;

```

TransverseBarCrossSectionArea : OPTIONAL IfcAreaMeasure;
 LongitudinalBarSpacing : OPTIONAL IfcPositiveLengthMeasure;
 TransverseBarSpacing : OPTIONAL IfcPositiveLengthMeasure;
 BendingShapeCode : OPTIONAL IfcLabel;
 BendingParameters : OPTIONAL LIST [1:?] OF IfcBendingParameterSelect;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcReinforcingMeshTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcReinforcingMeshTypeEnum.USERDEFINED) AND
 EXISTS(SELFWIfcElementType.ElementType));

BENDINGSHAPECODEPROVIDED : NOT EXISTS(BendingParameters) OR EXISTS(BendingShapeCode);

END_ENTITY;

ENTITY IfcTendonAnchorType

SUBTYPE OF(IfcReinforcingElementType);

PredefinedType : IfcTendonAnchorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTendonAnchorTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcTendonAnchorTypeEnum.USERDEFINED) AND
 EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcTendonType

SUBTYPE OF(IfcReinforcingElementType);

PredefinedType : IfcTendonTypeEnum;

NominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

CrossSectionArea : OPTIONAL IfcAreaMeasure;

SheethDiameter : OPTIONAL IfcPositiveLengthMeasure;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTendonTypeEnum.USERDEFINED) OR
 ((PredefinedType = IfcTendonTypeEnum.USERDEFINED) AND
 EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcGroundReinforcingElementType_K

SUBTYPE OF(IfcReinforcingElementType);

PredefinedType : IfcGroundReinforcingElementyTypeEnum_K;

END_ENTITY;

ENTITY IfcVibrationIsolatorType

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcVibrationIsolatorTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcVibrationIsolatorTypeEnum.USERDEFINED) OR

((PredefinedType = IfcVibrationIsolatorTypeEnum.USERDEFINED) AND

EXISTS(SELFWIfcElementType.ElementType));

END_ENTITY;

ENTITY IfcWaterProofingElementType_K

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcWaterProofElementTypeEnum_K;

END_ENTITY;

ENTITY IfcRoadElementPartType_K

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcRoadElementPartTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeElementPartType_K

SUBTYPE OF(IfcElementComponentType);

PredefinedType : IfcBridgeElementPartTypeEnum_K;

END_ENTITY;

```

ENTITY IfcTunnelElementPartType_K
  SUBTYPE OF(IfcElementComponentType);
  PredefinedType : IfcTunnelElementPartTypeEnum_K;
END_ENTITY;

```

```

ENTITY IfcFurnishingElementType
  SUPERTYPE OF (ONEOF(IfcFurnitureType, IfcSystemFurnitureElementType))
  SUBTYPE OF(IfcElementType);
END_ENTITY;

```

```

ENTITY IfcFurnitureType
  SUBTYPE OF(IfcFurnishingElementType);
  AssemblyPlace : IfcAssemblyPlaceEnum;
  PredefinedType : OPTIONAL IfcFurnitureTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcFurnitureTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcFurnitureTypeEnum.USERDEFINED) AND
    EXISTS(SELFIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcSystemFurnitureElementType
  SUBTYPE OF(IfcFurnishingElementType);
  PredefinedType : OPTIONAL IfcSystemFurnitureElementTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSystemFurnitureElementTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcSystemFurnitureElementTypeEnum.USERDEFINED)
    AND EXISTS(SELFIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcGeographicElementType
  SUBTYPE OF(IfcElementType);
  PredefinedType : IfcGeographicElementTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcGeographicElementTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcGeographicElementTypeEnum.USERDEFINED) AND
    EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcTransportElementType
  SUBTYPE OF(IfcElementType);
  PredefinedType : IfcTransportElementTypeEnum;
  WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcTransportElementTypeEnum.USERDEFINED) OR
      ((PredefinedType = IfcTransportElementTypeEnum.USERDEFINED) AND
    EXISTS(SELFWIfcElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcSpatialElementType
  ABSTRACT SUPERTYPE OF (ONEOF(IfcSpatialStructureElementType, IfcSpatialZoneType,
IfcCivilSpatialStructureElementType_K, IfcCivilSpatialBoundaryType_K))
  SUBTYPE OF(IfcTypeProduct);
  ElementType : OPTIONAL IfcLabel;
END_ENTITY;

```

```

ENTITY IfcSpatialStructureElementType
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcSpatialElementType);
END_ENTITY;

```

```

ENTITY IfcSpaceType

```

```

SUBTYPE OF(IfcSpatialStructureElementType);
    PredefinedType : IfcSpaceTypeEnum;
    LongName       : OPTIONAL IfcLabel;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSpaceTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcSpaceTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcSpatialElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcSpatialZoneType
    SUBTYPE OF(IfcSpatialElementType);
    PredefinedType : IfcSpatialZoneTypeEnum;
    LongName       : OPTIONAL IfcLabel;
WHERE
    CORRECTPREDEFINEDTYPE : (PredefinedType <> IfcSpatialZoneTypeEnum.USERDEFINED) OR
                            ((PredefinedType = IfcSpatialZoneTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcSpatialElementType.ElementType));
END_ENTITY;

```

```

ENTITY IfcCivilSpatialStructureElementType_K
    ABSTRACT SUPERTYPE OF (ONEOF(IfcRoadType_K, IfcBridgeType_K, IfcTunnelType_K))
    SUBTYPE OF(IfcSpatialElementType);
END_ENTITY;

```

```

ENTITY IfcRoadType_K
    SUBTYPE OF(IfcCivilSpatialStructureElementType_K);
    PredefinedType : IfcRoadTypeEnum_K;
END_ENTITY;

```

```

ENTITY IfcBridgeType_K

```

```
    SUBTYPE OF(IfcCivilSpatialStructureElementType_K);
    PredefinedType : IfcBridgeTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcTunnelType_K
    SUBTYPE OF(IfcCivilSpatialStructureElementType_K);
    PredefinedType : IfcTunnelTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcCivilSpatialBoundaryType_K
    ABSTRACT SUPERTYPE OF (ONEOF(IfcLinearRefSpaceType_K, IfcCurvilinearNodeSpaceType_K,
IfcVerticalSubspaceType_K))
    SUBTYPE OF(IfcSpatialElementType);
END_ENTITY;
```

```
ENTITY IfcLinearRefSpaceType_K
    SUBTYPE OF(IfcCivilSpatialBoundaryType_K);
    PredefinedType : IfcLinearRefSpaceTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcCurvilinearNodeSpaceType_K
    SUBTYPE OF(IfcCivilSpatialBoundaryType_K);
    PredefinedType : IfcCurvilinearNodeSpaceTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcVerticalSubspaceType_K
    SUBTYPE OF(IfcCivilSpatialBoundaryType_K);
    PredefinedType : IfcVerticalSubspaceTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcWindowStyle
  SUBTYPE OF(IfcTypeProduct);
  ConstructionType      : IfcWindowStyleConstructionEnum;
  OperationType        : IfcWindowStyleOperationEnum;
  ParameterTakesPrecedence : IfcBoolean;
  Sizeable              : IfcBoolean;
END_ENTITY;
```

```
ENTITY IfcCivilElement
  ABSTRACT SUPERTYPE OF (ONEOF(IfcSubsidiaryFacility_K, IfcEarthworkElement_K, IfcRoadElement_K,
IfcCivilStructureElement_K, IfcCivilElementProxy_K))
  SUBTYPE OF(IfcElement);
END_ENTITY;
```

```
ENTITY IfcSubsidiaryFacility_K
  ABSTRACT SUPERTYPE OF (ONEOF(IfcRoadSignEquipment_K, IfcGuard_K, IfcPavementAddition_K))
  SUBTYPE OF(IfcCivilElement);
END_ENTITY;
```

```
ENTITY IfcRoadSignEquipment_K
  SUBTYPE OF(IfcSubsidiaryFacility_K);
  PredefinedType : OPTIONAL IfcRoadSignEquipmentTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcGuard_K
  SUBTYPE OF(IfcSubsidiaryFacility_K);
  PredefinedType : OPTIONAL IfcGuardTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcPavementAddition_K
```

SUBTYPE OF(IfcSubsidiaryFacility_K);

PredefinedType : OPTIONAL IfcPavementAdditionTypeEnum_K;

END_ENTITY;

ENTITY IfcEarthworkElement_K

SUBTYPE OF(IfcCivilElement);

PredefinedType : OPTIONAL IfcEarthworkElementTypeEnum_K;

END_ENTITY;

ENTITY IfcRoadElement_K

ABSTRACT SUPERTYPE OF (ONEOF(IfcRoadBody_K, IfcRoadMedianStrip_K, IfcRoadShoulder_K, IfcCurb_K, IfcRoadPavement_K))

SUBTYPE OF(IfcCivilElement);

END_ENTITY;

ENTITY IfcRoadBody_K

SUBTYPE OF(IfcRoadElement_K);

PredefinedType : OPTIONAL IfcRoadBodyTypeEnum_K;

END_ENTITY;

ENTITY IfcRoadMedianStrip_K

SUBTYPE OF(IfcRoadElement_K);

PredefinedType : OPTIONAL IfcRoadMedianStripTypeEnum;

END_ENTITY;

ENTITY IfcRoadShoulder_K

SUBTYPE OF(IfcRoadElement_K);

PredefinedType : OPTIONAL IfcRoadShoulderTypeEnum_K;

END_ENTITY;

```
ENTITY IfcCurb_K
  SUBTYPE OF(IfcRoadElement_K);
  PredefinedType : OPTIONAL IfcCurbTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcRoadPavement_K
  SUBTYPE OF(IfcRoadElement_K);
  PredefinedType : OPTIONAL IfcPavementTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcCivilStructureElement_K
  ABSTRACT SUPERTYPE OF (ONEOF(IfcCulvert_K, IfcRetainingWall_K, IfcCaisson_K,
IfcBridgeElement_K, IfcTunnelElement_K))
  SUBTYPE OF(IfcCivilElement);
END_ENTITY;
```

```
ENTITY IfcCulvert_K
  SUBTYPE OF(IfcCivilStructureElement_K);
  PredefinedType : OPTIONAL IfcCulvertTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcRetainingWall_K
  SUBTYPE OF(IfcCivilStructureElement_K);
  PredefinedType : OPTIONAL IfcRetainingWallTypeEnum_K;
END_ENTITY;
```

```
ENTITY IfcCaisson_K
  SUBTYPE OF(IfcCivilStructureElement_K);
  PredefinedType : OPTIONAL IfcCaissonTypeEnum_K;
END_ENTITY;
```


ENTITY IfcBridgeElement_K

SUPERTYPE OF (ONEOF(IfcBridgeDeck_K, IfcBridgeTower_K, IfcBridgeCable_K, IfcBridgePier_K, IfcBridgeAbutment_K, IfcBridgeSpan_K, IfcBridgeSegment_K, IfcBridgeCoping_K, IfcBridgeGirder_K))

SUBTYPE OF(IfcCivilStructureElement_K);

END_ENTITY;

ENTITY IfcBridgeDeck_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeDeckTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeTower_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeTowerTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeCable_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeCableTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgePier_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgePierTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeAbutment_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeAbutmentTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeSpan_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeSpanTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeSegment_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeSegmentTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeCoping_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeCopingTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeGirder_K

SUBTYPE OF(IfcBridgeElement_K);

PredefinedType : OPTIONAL IfcBridgeGirderTypeEnum_K;

END_ENTITY;

ENTITY IfcTunnelElement_K

ABSTRACT SUPERTYPE OF (ONEOF(IfcTunnelLining_K, IfcTunnelLiningSegment_K))

SUBTYPE OF(IfcCivilStructureElement_K);

END_ENTITY;

ENTITY IfcTunnelLining_K

SUBTYPE OF(IfcTunnelElement_K);

PredefinedType : OPTIONAL IfcTunnelLiningTypeEnum_K;

END_ENTITY;

ENTITY IfcTunnelLiningSegment_K

SUBTYPE OF(IfcTunnelElement_K);

PredefinedType : OPTIONAL IfcTunnelLiningSegmentTypeEnum_K;

END_ENTITY;

ENTITY IfcCivilElementProxy_K

SUBTYPE OF(IfcCivilElement);

PredefinedType : OPTIONAL IfcCivilElementProxyTypeEnum_K;

END_ENTITY;

ENTITY IfcElementAssembly

SUBTYPE OF(IfcElement);

AssemblyPlace : OPTIONAL IfcAssemblyPlaceEnum;

PredefinedType : OPTIONAL IfcElementAssemblyTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcElementAssemblyTypeEnum.USERDEFINED) OR

((PredefinedType = IfcElementAssemblyTypeEnum.USERDEFINED) AND

EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCELEMENTASSEMBLYTYPE' IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcElementComponent

ABSTRACT SUPERTYPE OF (ONEOF(IfcBuildingElementPart, IfcDiscreteAccessory, IfcFastener, IfcMechanicalFastener, IfcReinforcingElement, IfcVibrationIsolator, IfcWaterProofingElement_K, IfcRoadElementPart_K, IfcBridgeElementPart_K, IfcTunnelElementPart_K))

SUBTYPE OF(IfcElement);

END_ENTITY;

ENTITY IfcBuildingElementPart

SUBTYPE OF(IfcElementComponent);

PredefinedType : OPTIONAL IfcBuildingElementPartTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcBuildingElementPartTypeEnum.USERDEFINED) OR

((PredefinedType = IfcBuildingElementPartTypeEnum.USERDEFINED) AND

EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCBUILDINGELEMENTPARTTYPE' IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcDiscreteAccessory

SUBTYPE OF(IfcElementComponent);

PredefinedType : OPTIONAL IfcDiscreteAccessoryTypeEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

(PredefinedType <> IfcDiscreteAccessoryTypeEnum.USERDEFINED) OR

((PredefinedType = IfcDiscreteAccessoryTypeEnum.USERDEFINED) AND

EXISTS (SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCDISCRETEACCESSORYTYPE' IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcFastener

SUBTYPE OF(IfcElementComponent);

PredefinedType : OPTIONAL IfcFastenerTypeEnum;

WHERE

```
CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcFastenerTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcFastenerTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCFASTENERTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

ENTITY IfcMechanicalFastener

SUBTYPE OF(IfcElementComponent);

NominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

NominalLength : OPTIONAL IfcPositiveLengthMeasure;

PredefinedType : OPTIONAL IfcMechanicalFastenerTypeEnum;

WHERE

```
CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcMechanicalFastenerTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcMechanicalFastenerTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED   : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCMECHANICALFASTENERTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

ENTITY IfcReinforcingElement

ABSTRACT SUPERTYPE OF (ONEOF(IfcReinforcingBar, IfcReinforcingMesh, IfcTendon, IfcTendonAnchor)
ANDOR IfcGroundReinforcingElement_K)

SUBTYPE OF(IfcElementComponent);

SteelGrade : OPTIONAL IfcLabel;

END_ENTITY;

ENTITY IfcReinforcingBar

SUBTYPE OF(IfcReinforcingElement);

NominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

CrossSectionArea : OPTIONAL IfcAreaMeasure;

BarLength : OPTIONAL IfcPositiveLengthMeasure;

PredefinedType : OPTIONAL IfcReinforcingBarTypeEnum;

BarSurface : OPTIONAL IfcReinforcingBarSurfaceEnum;

WHERE

CORRECTPREDEFINEDTYPE : NOT EXISTS(PredefinedType) OR

(PredefinedType <> IfcReinforcingBarTypeEnum.USERDEFINED) OR

((PredefinedType = IfcReinforcingBarTypeEnum.USERDEFINED) AND

EXISTS(SELFIfcObject.ObjectType));

CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

('IFC4.IFCREINFORCINGBARTYPE' IN

TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

END_ENTITY;

ENTITY IfcReinforcingMesh

SUBTYPE OF(IfcReinforcingElement);

MeshLength : OPTIONAL IfcPositiveLengthMeasure;

MeshWidth : OPTIONAL IfcPositiveLengthMeasure;

LongitudinalBarNominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

TransverseBarNominalDiameter : OPTIONAL IfcPositiveLengthMeasure;

LongitudinalBarCrossSectionArea : OPTIONAL IfcAreaMeasure;

TransverseBarCrossSectionArea : OPTIONAL IfcAreaMeasure;

LongitudinalBarSpacing : OPTIONAL IfcPositiveLengthMeasure;

TransverseBarSpacing : OPTIONAL IfcPositiveLengthMeasure;

PredefinedType : OPTIONAL IfcReinforcingMeshTypeEnum;

WHERE

```

CORRECTPREDEFINEDTYPE : NOT EXISTS(PredefinedType) OR
                        (PredefinedType <> IfcReinforcingMeshTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcReinforcingMeshTypeEnum.USERDEFINED) AND
EXISTS(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED  : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCREINFORCINGMESHTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcTendon

SUBTYPE OF(IfcReinforcingElement);

```

PredefinedType      : OPTIONAL IfcTendonTypeEnum;
NominalDiameter     : OPTIONAL IfcPositiveLengthMeasure;
CrossSectionArea    : OPTIONAL IfcAreaMeasure;
TensionForce        : OPTIONAL IfcForceMeasure;
PreStress           : OPTIONAL IfcPressureMeasure;
FrictionCoefficient : OPTIONAL IfcNormalisedRatioMeasure;
AnchorageSlip      : OPTIONAL IfcPositiveLengthMeasure;
MinCurvatureRadius : OPTIONAL IfcPositiveLengthMeasure;

```

WHERE

```

CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                        (PredefinedType <> IfcTendonTypeEnum.USERDEFINED) OR
                        ((PredefinedType = IfcTendonTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
CORRECTTYPEASSIGNED  : (SIZEOF(IsTypedBy) = 0) OR
                        ('IFC4.IFCTENDONTYPE'
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

ENTITY IfcTendonAnchor

SUBTYPE OF(IfcReinforcingElement);

```

    PredefinedType : OPTIONAL IfcTendonAnchorTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcTendonAnchorTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcTendonAnchorTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCTENDONANCHORTYPE'
                                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcGroundReinforcingElement_K
    SUBTYPE OF(IfcReinforcingElement);
    PredefinedType : OPTIONAL IfcGroundReinforcingElementTypeEnum_K;
END_ENTITY;

```

```

ENTITY IfcVibrationIsolator
    SUBTYPE OF(IfcElementComponent);
    PredefinedType : OPTIONAL IfcVibrationIsolatorTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcVibrationIsolatorTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcVibrationIsolatorTypeEnum.USERDEFINED) AND
EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCVIBRATIONISOLATORTYPE'
                                IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcWaterProofingElement_K
    SUBTYPE OF(IfcElementComponent);

```


PredefinedType : OPTIONAL IfcWaterProofElementTypeEnum_K;
END_ENTITY;

ENTITY IfcRoadElementPart_K

SUBTYPE OF(IfcElementComponent);

PredefinedType : OPTIONAL IfcRoadElementPartTypeEnum_K;

END_ENTITY;

ENTITY IfcBridgeElementPart_K

SUBTYPE OF(IfcElementComponent);

PredefinedType : OPTIONAL IfcBridgeElementPartTypeEnum_K;

END_ENTITY;

ENTITY IfcTunnelElementPart_K

SUBTYPE OF(IfcElementComponent);

PredefinedType : OPTIONAL IfcTunnelElementPartTypeEnum_K;

END_ENTITY;

ENTITY IfcFeatureElement

ABSTRACT SUPERTYPE OF (ONEOF(IfcFeatureElementAddition, IfcFeatureElementSubtraction,
IfcSurfaceFeature))

SUBTYPE OF(IfcElement);

END_ENTITY;

ENTITY IfcFeatureElementAddition

ABSTRACT SUPERTYPE

SUBTYPE OF(IfcFeatureElement);

INVERSE

ProjectsElements : IfcRelProjectsElement FOR RelatedFeatureElement;

END_ENTITY;

```
ENTITY IfcProjectionElement
  SUBTYPE OF(IfcFeatureElementAddition);
  PredefinedType : OPTIONAL IfcProjectionElementTypeEnum;
END_ENTITY;
```

```
ENTITY IfcRelProjectsElement
  SUBTYPE OF(IfcRelDecomposes);
  RelatingElement : IfcElement;
  RelatedFeatureElement : IfcFeatureElementAddition;
END_ENTITY;
```

```
ENTITY IfcRelDecomposes
  ABSTRACT SUPERTYPE OF (ONEOF(IfcRelAggregates, IfcRelNests, IfcRelProjectsElement,
IfcRelVoidsElement))
  SUBTYPE OF(IfcRelationship);
END_ENTITY;
```

```
ENTITY IfcRelAggregates
  SUBTYPE OF(IfcRelDecomposes);
  RelatingObject : IfcObjectDefinition;
  RelatedObjects : SET [1:?] OF IfcObjectDefinition;
  WHERE
    NOSELFREFERENCE : SIZEOF(QUERY(Temp <* RelatedObjects | RelatingObject :=: Temp)) = 0;
END_ENTITY;
```

```
ENTITY IfcRelNests
  SUBTYPE OF(IfcRelDecomposes);
  RelatingObject : IfcObjectDefinition;
  RelatedObjects : LIST [1:?] OF IfcObjectDefinition;
```

```

WHERE
    NOSELFREFERENCE : SIZEOF(QUERY(Temp <* RelatedObjects | RelatingObject :=: Temp)) = 0;
END_ENTITY;

ENTITY IfcRelVoidsElement
    SUBTYPE OF(IfcRelDecomposes);
    RelatingBuildingElement : IfcElement;
    RelatedOpeningElement : IfcFeatureElementSubtraction;
END_ENTITY;

ENTITY IfcFeatureElementSubtraction
    ABSTRACT SUPERTYPE OF (ONEOF(IfcOpeningElement, IfcVoidingFeature))
    SUBTYPE OF(IfcFeatureElement);
    INVERSE
        VoidsElements : IfcRelVoidsElement FOR RelatedOpeningElement;
    WHERE
        HASNOSUBTRACTION : SIZEOF(SELFWIfcElement.HasOpenings) = 0;
        ISNOTFILLING : SIZEOF(SELFWIfcElement.FillsVoids) = 0;
END_ENTITY;

ENTITY IfcOpeningElement
    SUBTYPE OF(IfcFeatureElementSubtraction);
    PredefinedType : OPTIONAL IfcOpeningElementTypeEnum;
    INVERSE
        HasFillings : SET OF IfcRelFillsElement FOR RelatingOpeningElement;
END_ENTITY;

ENTITY IfcOpeningStandardCase
    SUBTYPE OF(IfcOpeningElement);
END_ENTITY;

```

ENTITY IfcReIFillsElement

SUBTYPE OF(IfcReIConnects);

RelatingOpeningElement : IfcOpeningElement;

RelatedBuildingElement : IfcElement;

END_ENTITY;

ENTITY IfcVoidingFeature

SUBTYPE OF(IfcFeatureElementSubtraction);

PredefinedType : OPTIONAL IfcVoidingFeatureTypeEnum;

WHERE

HASOBJECTTYPE : NOT EXISTS(PredefinedType) OR (PredefinedType <>
IfcVoidingFeatureTypeEnum.USERDEFINED) OR EXISTS(SELFWIfcObject.ObjectType);

END_ENTITY;

ENTITY IfcSurfaceFeature

SUBTYPE OF(IfcFeatureElement);

PredefinedType : OPTIONAL IfcSurfaceFeatureTypeEnum;

WHERE

HASOBJECTTYPE : NOT EXISTS(PredefinedType) OR (PredefinedType <>
IfcSurfaceFeatureTypeEnum.USERDEFINED) OR EXISTS(SELFWIfcObject.ObjectType);

END_ENTITY;

ENTITY IfcFurnishingElement

SUPERTYPE OF (ONEOF(IfcFurniture, IfcSystemFurnitureElement))

SUBTYPE OF(IfcElement);

END_ENTITY;

ENTITY IfcFurniture

SUBTYPE OF(IfcFurnishingElement);

```

    PredefinedType : OPTIONAL IfcFurnitureTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcFurnitureTypeEnum.USERDEFINED) OR
                                ((PredefinedType = IfcFurnitureTypeEnum.USERDEFINED) AND EXISTS
(SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCFURNITURETYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcSystemFurnitureElement
    SUBTYPE OF(IfcFurnishingElement);
    PredefinedType : OPTIONAL IfcSystemFurnitureElementTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcSystemFurnitureElementTypeEnum.USERDEFINED)
OR
                                ((PredefinedType = IfcSystemFurnitureElementTypeEnum.USERDEFINED)
AND EXISTS (SELFWIfcObject.ObjectType));
    CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                                ('IFC4.IFCSYSTEMFURNITUREELEMENTTYPE' IN
TYPEOF(SELFWIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcGeographicElement
    SUBTYPE OF(IfcElement);
    PredefinedType : OPTIONAL IfcGeographicElementTypeEnum;
WHERE
    CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR
                                (PredefinedType <> IfcGeographicElementTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcGeographicElementTypeEnum.USERDEFINED) AND
EXISTS (SELFIfcObject.ObjectType));
        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR
                ('IFC4.IFCGEOGRAPHICELEMENTTYPE'
IN
TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;

```

```

ENTITY IfcTransportElement

```

```

    SUBTYPE OF(IfcElement);

```

```

    PredefinedType : OPTIONAL IfcTransportElementTypeEnum;

```

```

    WHERE

```

```

        CORRECTPREDEFINEDTYPE : NOT(EXISTS(PredefinedType)) OR

```

```

                (PredefinedType <> IfcTransportElementTypeEnum.USERDEFINED) OR

```

```

                ((PredefinedType = IfcTransportElementTypeEnum.USERDEFINED) AND
EXISTS (SELFIfcObject.ObjectType));

```

```

        CORRECTTYPEASSIGNED : (SIZEOF(IsTypedBy) = 0) OR

```

```

                ('IFC4.IFCTRANSPORTELEMENTTYPE'
IN
TYPEOF(SELFIfcObject.IsTypedBy[1].RelatingType));

```

```

END_ENTITY;

```

```

ENTITY IfcVirtualElement

```

```

    SUBTYPE OF(IfcElement);

```

```

END_ENTITY;

```

```

ENTITY IfcRelInterferesElements

```

```

    SUBTYPE OF(IfcRelConnects);

```

```

    RelatingElement : IfcElement;

```

```

    RelatedElement : IfcElement;

```

```

    InterferenceGeometry : OPTIONAL IfcConnectionGeometry;

```

```

    InterferenceType : OPTIONAL IfcIdentifier;

```

```

    ImpliedOrder : LOGICAL;

```

WHERE

NOTSELFREFERENCE : RelatingElement :<>: RelatedElement;

END_ENTITY;

ENTITY IfcRelAssignsToGroup

SUBTYPE OF(IfcRelAssigns);

RelatingGroup : IfcGroup;

WHERE

NOSELFREFERENCE : SIZEOF(QUERY(Temp <* SELFWIfcRelAssigns.RelatedObjects | RelatingGroup :=: Temp)) = 0;

END_ENTITY;

ENTITY IfcRelAssignsToGroupByFactor

SUBTYPE OF(IfcRelAssignsToGroup);

Factor : IfcRatioMeasure;

END_ENTITY;

ENTITY IfcOccupant

SUBTYPE OF(IfcActor);

PredefinedType : OPTIONAL IfcOccupantTypeEnum;

WHERE

WR31 : NOT(PredefinedType = IfcOccupantTypeEnum.USERDEFINED)

OR EXISTS(SELFWIfcObject.ObjectType);

END_ENTITY;

ENTITY IfcRelDeclares

SUBTYPE OF(IfcRelationship);

RelatingContext : IfcContext;

RelatedDefinitions : SET [1:?] OF IfcDefinitionSelect;

WHERE

```
NOSELFREFERENCE : SIZEOF(QUERY(Temp <* RelatedDefinitions | RelatingContext :=: Temp)) = 0;
END_ENTITY;
```

```
ENTITY IfcRelAssociatesConstraint
  SUBTYPE OF(IfcRelAssociates);
  Intent : OPTIONAL IfcLabel;
  RelatingConstraint : IfcConstraint;
END_ENTITY;
```

```
ENTITY IfcRelAssociatesMaterial
  SUBTYPE OF(IfcRelAssociates);
  RelatingMaterial : IfcMaterialSelect;
  WHERE
    NOVOIDELEMENT : SIZEOF(QUERY(temp <* SELFWIfcRelAssociates.RelatedObjects |
      ('IFC4.IFCFEATUREELEMENTSUBTRACTION' IN TYPEOF(temp)) OR
      ('IFC4.IFCVIRTUALELEMENT' IN TYPEOF(temp))
    )) = 0;
    ALLOWEDELEMENTS : SIZEOF(QUERY(temp <* SELFWIfcRelAssociates.RelatedObjects | (
      SIZEOF(TYPEOF(temp) * [
        'IFC4.IFCELEMENT',
        'IFC4.IFCELEMENTTYPE',
        'IFC4.IFCWINDOWSTYLE',
        'IFC4.IFCDOORSTYLE',
        'IFC4.IFCSTRUCTURALMEMBER',
        'IFC4.IFCPORT']) = 0)
    )) = 0;
END_ENTITY;
```

```
ENTITY IfcMaterialList;
  Materials : LIST [1:?] OF IfcMaterial;
```


END_ENTITY;

ENTITY IfcMaterialUsageDefinition

ABSTRACT SUPERTYPE OF (ONEOF(IfcMaterialLayerSetUsage, IfcMaterialProfileSetUsage));

INVERSE

AssociatedTo : SET [1:?] OF IfcRelAssociatesMaterial FOR RelatingMaterial;

END_ENTITY;

ENTITY IfcMaterialLayerSetUsage

SUBTYPE OF(IfcMaterialUsageDefinition);

ForLayerSet : IfcMaterialLayerSet;

LayerSetDirection : IfcLayerSetDirectionEnum;

DirectionSense : IfcDirectionSenseEnum;

OffsetFromReferenceLine : IfcLengthMeasure;

ReferenceExtent : OPTIONAL IfcPositiveLengthMeasure;

END_ENTITY;

ENTITY IfcMaterialProfileSetUsage

SUBTYPE OF(IfcMaterialUsageDefinition);

ForProfileSet : IfcMaterialProfileSet;

CardinalPoint : OPTIONAL IfcCardinalPointReference;

ReferenceExtent : OPTIONAL IfcPositiveLengthMeasure;

END_ENTITY;

ENTITY IfcMaterialProfileSetUsageTapering

SUBTYPE OF(IfcMaterialProfileSetUsage);

ForProfileEndSet : IfcMaterialProfileSet;

CardinalEndPoint : OPTIONAL IfcCardinalPointReference;

END_ENTITY;

```
ENTITY IfcMaterialProfileSet
  SUBTYPE OF(IfcMaterialDefinition);
  Name          : OPTIONAL IfcLabel;
  Description    : OPTIONAL IfcText;
  MaterialProfiles : LIST [1:?] OF IfcMaterialProfile;
  CompositeProfile : OPTIONAL IfcCompositeProfileDef;
END_ENTITY;
```

```
ENTITY IfcCurrencyRelationship
  SUBTYPE OF(IfcResourceLevelRelationship);
  RelatingMonetaryUnit : IfcMonetaryUnit;
  RelatedMonetaryUnit  : IfcMonetaryUnit;
  ExchangeRate         : IfcPositiveRatioMeasure;
  RateDateTime         : OPTIONAL IfcDateTime;
  RateSource           : OPTIONAL IfcLibraryInformation;
END_ENTITY;
```

```
ENTITY IfcMaterialRelationship
  SUBTYPE OF(IfcResourceLevelRelationship);
  RelatingMaterial : IfcMaterial;
  RelatedMaterials : SET [1:?] OF IfcMaterial;
  Expression       : OPTIONAL IfcLabel;
END_ENTITY;
```

```
ENTITY IfcStyleModel
  ABSTRACT SUPERTYPE
  SUBTYPE OF(IfcRepresentation);
END_ENTITY;
```

```
ENTITY IfcStyledRepresentation
```

```

SUBTYPE OF(IfcStyleModel);
WHERE
    ONLYSTYLEDITEMS : SIZEOF(QUERY(temp <* SELFIfcRepresentation.Items |
        (NOT('IFC4.IFCSTYLEDITEM' IN TYPEOF(temp)))
        )) = 0;
END_ENTITY;

```

```

ENTITY IfcPresentationLayerAssignment;
    Name          : IfcLabel;
    Description    : OPTIONAL IfcText;
    AssignedItems : SET [1:?] OF IfcLayeredItem;
    Identifier     : OPTIONAL IfcIdentifier;
WHERE
    APPLICABLEITEMS : SIZEOF(QUERY(temp <* AssignedItems | (
        SIZEOF(TYPEOF(temp) * [
            'IFC4.IFCSHAPE REPRESENTATION',
            'IFC4.IFCGEOMETRICREPRESENTATIONITEM',
            'IFC4.IFCMAPPEDITEM']) = 1)
        )) = SIZEOF(AssignedItems);
END_ENTITY;

```

```

ENTITY IfcPresentationLayerWithStyle
SUBTYPE OF(IfcPresentationLayerAssignment);
    LayerOn       : IfcLogical;
    LayerFrozen   : IfcLogical;
    LayerBlocked  : IfcLogical;
    LayerStyles   : SET OF IfcPresentationStyle;
WHERE
    APPLICABLEONLYTOITEMS : SIZEOF(QUERY(temp <* AssignedItems | (
        SIZEOF(TYPEOF(temp) * [

```

```

        'IFC4.IFCGEOMETRICREPRESENTATIONITEM',
        'IFC4.IFCMAPPEDITEM']) = 1)
    )) = SIZEOF(AssignedItems);

```

END_ENTITY;

ENTITY IfcPresentationStyle

ABSTRACT SUPERTYPE OF (ONEOF(IfcCurveStyle, IfcFillAreaStyle, IfcSurfaceStyle, IfcTextStyle));

Name : OPTIONAL IfcLabel;

END_ENTITY;

ENTITY IfcCurveStyle

SUBTYPE OF(IfcPresentationStyle);

CurveFont : OPTIONAL IfcCurveFontOrScaledCurveFontSelect;

CurveWidth : OPTIONAL IfcSizeSelect;

CurveColour : OPTIONAL IfcColour;

ModelOrDraughting : OPTIONAL IfcBoolean;

WHERE

MEASUREOFWIDTH : (NOT(EXISTS(CurveWidth))) OR
('IFC4.IFCPOSITIVELENGTHMEASURE' IN TYPEOF(CurveWidth)) OR
(('IFC4.IFCDESCRIPTIVEMEASURE' IN TYPEOF(CurveWidth)) AND
(CurveWidth = 'by layer'));

IDENTIFIABLECURVESTYLE : EXISTS(CurveFont) OR EXISTS(CurveWidth) OR EXISTS(CurveColour);

END_ENTITY;

ENTITY IfcFillAreaStyle

SUBTYPE OF(IfcPresentationStyle);

FillStyles : SET [1:?] OF IfcFillStyleSelect;

ModelOrDraughting : OPTIONAL IfcBoolean;

WHERE

MAXONECOLOUR : SIZEOF(QUERY(Style <* SELF.FillStyles |

```

        'IFC4.IFCCOLOUR' IN
        TYPEOF(Style)
    )) <= 1;
MAXONEEXTHATCHSTYLE : SIZEOF(QUERY(Style <* SELF.FillStyles |
        'IFC4.IFCEXTERNALLYDEFINEDHATCHSTYLE' IN
        TYPEOF(Style)
    )) <= 1;
CONSISTENTHATCHSTYLEDEF : IfcCorrectFillAreaStyle(SELF.FillStyles);
END_ENTITY;

```

ENTITY IfcFillAreaStyleHatching

SUBTYPE OF(IfcGeometricRepresentationItem);

```

HatchLineAppearance : IfcCurveStyle;
StartOfNextHatchLine : IfcHatchLineDistanceSelect;
PointOfReferenceHatchLine : OPTIONAL IfcCartesianPoint;
PatternStart : OPTIONAL IfcCartesianPoint;
HatchLineAngle : IfcPlaneAngleMeasure;

```

WHERE

PATTERNSTART2D : NOT(EXISTS(PatternStart)) OR (PatternStart.Dim = 2);

REFHATCHLINE2D : NOT(EXISTS(PointOfReferenceHatchLine)) OR (PointOfReferenceHatchLine.Dim = 2);

END_ENTITY;

ENTITY IfcFillAreaStyleTiles

SUBTYPE OF(IfcGeometricRepresentationItem);

```

TilingPattern : LIST [2:2] OF IfcVector;
Tiles : SET [1:?] OF IfcStyledItem;
TilingScale : IfcPositiveRatioMeasure;

```

END_ENTITY;

```

ENTITY IfcStyledItem
  SUBTYPE OF(IfcRepresentationItem);
  Item    : OPTIONAL IfcRepresentationItem;
  Styles  : SET [1:?] OF IfcStyleAssignmentSelect;
  Name    : OPTIONAL IfcLabel;
  WHERE
    APPLICABLEITEM : NOT('IFC4.IFCSTYLEDITEM' IN TYPEOF(Item));
END_ENTITY;

```

```

ENTITY IfcPresentationStyleAssignment;
  Styles : SET [1:?] OF IfcPresentationStyleSelect;
END_ENTITY;

```

```

ENTITY IfcSurfaceStyle
  SUBTYPE OF(IfcPresentationStyle);
  Side    : IfcSurfaceSide;
  Styles  : SET [1:5] OF IfcSurfaceStyleElementSelect;
  WHERE
    MAXONESHADING    : SIZEOF(QUERY(Style <* SELF.Styles |
                                   'IFC4.IFCSURFACESTYLESHADING' IN
                                   TYPEOF(Style)
                                   )) <= 1;
    MAXONELIGHTING   : SIZEOF(QUERY(Style <* SELF.Styles |
                                   'IFC4.IFCSURFACESTYLELIGHTING' IN
                                   TYPEOF(Style)
                                   )) <= 1;
    MAXONEREFRACTION : SIZEOF(QUERY(Style <* SELF.Styles |
                                   'IFC4.IFCSURFACESTYLEREFRACTION' IN
                                   TYPEOF(Style)
                                   )) <= 1;

```

```

MAXONETEXTURES : SIZEOF(QUERY(Style <* SELF.Styles |
                        'IFC4.IFCSURFACESTYLEWITHTEXTURES' IN
                        TYPEOF(Style)
                        )) <= 1;
MAXONEEXTDEFINED : SIZEOF(QUERY(Style <* SELF.Styles |
                        'IFC4.IFCEXTERNALLYDEFINEDSURFACESTYLE' IN
                        TYPEOF(Style)
                        )) <= 1;

```

END_ENTITY;

ENTITY IfcTextStyle

```

SUBTYPE OF(IfcPresentationStyle);
    TextCharacterAppearance : OPTIONAL IfcTextStyleForDefinedFont;
    TextStyle                : OPTIONAL IfcTextStyleTextModel;
    TextFontStyle           : IfcTextFontSelect;
    ModelOrDrafting         : OPTIONAL IfcBoolean;

```

END_ENTITY;

ENTITY IfcAnnotationFillArea

```

SUBTYPE OF(IfcGeometricRepresentationItem);
    OuterBoundary : IfcCurve;
    InnerBoundaries : OPTIONAL SET [1:?] OF IfcCurve;

```

END_ENTITY;

ENTITY IfcGeometricSet

```

SUBTYPE OF(IfcGeometricRepresentationItem);
    Elements : SET [1:?] OF IfcGeometricSetSelect;
DERIVE
    Dim : IfcDimensionCount := Elements[1].Dim;

```

WHERE

```
CONSISTENTDIM : SIZEOF(QUERY(Temp <* Elements |
    Temp.Dim <> Elements[1].Dim))
    = 0;
```

```
END_ENTITY;
```

```
ENTITY IfcGeometricCurveSet
```

```
    SUBTYPE OF(IfcGeometricSet);
```

```
    WHERE
```

```
        NOSURFACES : SIZEOF(QUERY(Temp <* SELFIfcGeometricSet.Elements |
            'IFC4.IFCSURFACE' IN TYPEOF(Temp))) = 0;
```

```
END_ENTITY;
```

```
ENTITY IfcLightSource
```

```
    ABSTRACT SUPERTYPE OF (ONEOF(IfcLightSourceAmbient, IfcLightSourceDirectional,
    IfcLightSourceGoniometric, IfcLightSourcePositional))
```

```
    SUBTYPE OF(IfcGeometricRepresentationItem);
```

```
        Name : OPTIONAL IfcLabel;
```

```
        LightColour : IfcColourRgb;
```

```
        AmbientIntensity : OPTIONAL IfcNormalisedRatioMeasure;
```

```
        Intensity : OPTIONAL IfcNormalisedRatioMeasure;
```

```
END_ENTITY;
```

```
ENTITY IfcLightSourceAmbient
```

```
    SUBTYPE OF(IfcLightSource);
```

```
END_ENTITY;
```

```
ENTITY IfcLightSourceDirectional
```

```
    SUBTYPE OF(IfcLightSource);
```

```
        Orientation : IfcDirection;
```

```
END_ENTITY;
```



```

ENTITY IfcLightSourceGoniometric
  SUBTYPE OF(IfcLightSource);
    Position                : IfcAxis2Placement3D;
    ColourAppearance       : OPTIONAL IfcColourRgb;
    ColourTemperature      : IfcThermodynamicTemperatureMeasure;
    LuminousFlux           : IfcLuminousFluxMeasure;
    LightEmissionSource    : IfcLightEmissionSourceEnum;
    LightDistributionDataSource : IfcLightDistributionDataSourceSelect;
END_ENTITY;

```

```

ENTITY IfcLightIntensityDistribution;
  LightDistributionCurve : IfcLightDistributionCurveEnum;
  DistributionData       : LIST [1:?] OF IfcLightDistributionData;
END_ENTITY;

```

```

ENTITY IfcLightDistributionData;
  MainPlaneAngle      : IfcPlaneAngleMeasure;
  SecondaryPlaneAngle : LIST [1:?] OF IfcPlaneAngleMeasure;
  LuminousIntensity   : LIST [1:?] OF IfcLuminousIntensityDistributionMeasure;
END_ENTITY;

```

```

ENTITY IfcLightSourcePositional
  SUBTYPE OF(IfcLightSource);
    Position          : IfcCartesianPoint;
    Radius            : IfcPositiveLengthMeasure;
    ConstantAttenuation : IfcReal;
    DistanceAttenuation : IfcReal;
    QuadraticAttenuation : IfcReal;
END_ENTITY;

```

ENTITY IfcLightSourceSpot

SUBTYPE OF(IfcLightSourcePositional);

Orientation : IfcDirection;

ConcentrationExponent : OPTIONAL IfcReal;

SpreadAngle : IfcPositivePlaneAngleMeasure;

BeamWidthAngle : IfcPositivePlaneAngleMeasure;

END_ENTITY;

ENTITY IfcPlanarExtent

SUBTYPE OF(IfcGeometricRepresentationItem);

SizeInX : IfcLengthMeasure;

SizeInY : IfcLengthMeasure;

END_ENTITY;

ENTITY IfcPlanarBox

SUBTYPE OF(IfcPlanarExtent);

Placement : IfcAxis2Placement;

END_ENTITY;

ENTITY IfcSectionedSpine

SUBTYPE OF(IfcGeometricRepresentationItem);

SpineCurve : IfcCompositeCurve;

CrossSections : LIST [2:?] OF IfcProfileDef;

CrossSectionPositions : LIST [2:?] OF IfcAxis2Placement3D;

DERIVE

Dim : IfcDimensionCount := 3;

WHERE

CORRESPONDINGSECTIONPOSITIONS : SIZEOF(CrossSections) = SIZEOF(CrossSectionPositions);

CONSISTENTPROFILETYPES : SIZEOF(QUERY(temp <* CrossSections |

```
CrossSections[1].ProfileType <> temp.ProfileType)) = 0;  
    SPINECURVEDIM          : SpineCurve.Dim = 3;  
END_ENTITY;
```

```
ENTITY IfcShellBasedSurfaceModel  
    SUBTYPE OF(IfcGeometricRepresentationItem);  
    SbsmBoundary : SET [1:?] OF IfcShell;  
    DERIVE  
    Dim          : IfcDimensionCount := 3;  
END_ENTITY;
```

```
ENTITY IfcTextLiteral  
    SUBTYPE OF(IfcGeometricRepresentationItem);  
    Literal      : IfcPresentableText;  
    Placement    : IfcAxis2Placement;  
    Path         : IfcTextPath;  
END_ENTITY;
```

```
ENTITY IfcTextLiteralWithExtent  
    SUBTYPE OF(IfcTextLiteral);  
    Extent       : IfcPlanarExtent;  
    BoxAlignment : IfcBoxAlignment;  
    WHERE  
    WR31 : NOT('IFC4.IFCPLANARBOX' IN TYPEOF(Extent));  
END_ENTITY;
```

```
ENTITY IfcCoordinateOperation  
    ABSTRACT SUPERTYPE;  
    SourceCRS : IfcCoordinateReferenceSystemSelect;  
    TargetCRS : IfcCoordinateReferenceSystem;
```

END_ENTITY;

ENTITY IfcMapConversion

SUBTYPE OF(IfcCoordinateOperation);

Eastings : IfcLengthMeasure;

Northings : IfcLengthMeasure;

OrthogonalHeight : IfcLengthMeasure;

XAxisAbscissa : OPTIONAL IfcReal;

XAxisOrdinate : OPTIONAL IfcReal;

Scale : OPTIONAL IfcReal;

END_ENTITY;

ENTITY IfcCoordinateReferenceSystem

ABSTRACT SUPERTYPE;

Name : IfcLabel;

Description : OPTIONAL IfcText;

GeodeticDatum : OPTIONAL IfcIdentifier;

VerticalDatum : OPTIONAL IfcIdentifier;

INVERSE

HasCoordinateOperation : SET [0:1] OF IfcCoordinateOperation FOR SourceCRS;

END_ENTITY;

ENTITY IfcProjectedCRS

SUBTYPE OF(IfcCoordinateReferenceSystem);

MapProjection : OPTIONAL IfcIdentifier;

MapZone : OPTIONAL IfcIdentifier;

MapUnit : OPTIONAL IfcNamedUnit;

WHERE

ISLENGTHUNIT : NOT(EXISTS(MapUnit)) OR (MapUnit.UnitType = IfcUnitEnum.LENGTHUNIT);

END_ENTITY;

ENTITY IfcUnitAssignment;

Units : SET [1:?] OF IfcUnit;

WHERE

WR01 : IfcCorrectUnitAssignment(Units);

END_ENTITY;

ENTITY IfcOwnerHistory;

OwningUser : IfcPersonAndOrganization;

OwningApplication : IfcApplication;

State : OPTIONAL IfcStateEnum;

ChangeAction : OPTIONAL IfcChangeActionEnum;

LastModifiedDate : OPTIONAL IfcTimeStamp;

LastModifyingUser : OPTIONAL IfcPersonAndOrganization;

LastModifyingApplication : OPTIONAL IfcApplication;

CreationDate : IfcTimeStamp;

WHERE

CORRECTCHANGEACTION : (EXISTS>LastModifiedDate)) OR

(NOT(EXISTS>LastModifiedDate)) AND NOT(EXISTS(ChangeAction))) OR

(NOT(EXISTS>LastModifiedDate)) AND EXISTS(ChangeAction) AND

((ChangeAction = IfcChangeActionEnum.NOTDEFINED) OR (ChangeAction = IfcChangeActionEnum.NOCHANGE)));

END_ENTITY;

ENTITY IfcApplication;

ApplicationDeveloper : IfcOrganization;

Version : IfcLabel;

ApplicationFullName : IfcLabel;

ApplicationIdentifier : IfcIdentifier;

UNIQUE

UR1 : ApplicationIdentifier;

UR2 : ApplicationFullName, Version;

END_ENTITY;

ENTITY IfcPreDefinedPropertySet

ABSTRACT SUPERTYPE OF (ONEOF(IfcDoorLiningProperties, IfcDoorPanelProperties, IfcPermeableCoveringProperties, IfcReinforcementDefinitionProperties, IfcWindowLiningProperties, IfcWindowPanelProperties))

SUBTYPE OF(IfcPropertySetDefinition);

END_ENTITY;

ENTITY IfcDoorLiningProperties

SUBTYPE OF(IfcPreDefinedPropertySet);

LiningDepth : OPTIONAL IfcPositiveLengthMeasure;

LiningThickness : OPTIONAL IfcNonNegativeLengthMeasure;

ThresholdDepth : OPTIONAL IfcPositiveLengthMeasure;

ThresholdThickness : OPTIONAL IfcNonNegativeLengthMeasure;

TransomThickness : OPTIONAL IfcNonNegativeLengthMeasure;

TransomOffset : OPTIONAL IfcLengthMeasure;

LiningOffset : OPTIONAL IfcLengthMeasure;

ThresholdOffset : OPTIONAL IfcLengthMeasure;

CasingThickness : OPTIONAL IfcPositiveLengthMeasure;

CasingDepth : OPTIONAL IfcPositiveLengthMeasure;

ShapeAspectStyle : OPTIONAL IfcShapeAspect;

LiningToPanelOffsetX : OPTIONAL IfcLengthMeasure;

LiningToPanelOffsetY : OPTIONAL IfcLengthMeasure;

WHERE

WR31 : NOT(EXISTS(LiningDepth) AND NOT(EXISTS(LiningThickness)));

WR32 : NOT(EXISTS(ThresholdDepth) AND NOT(EXISTS(ThresholdThickness)));

WR33 : (EXISTS(TransomOffset) AND EXISTS(TransomThickness)) XOR

```

        (NOT(EXISTS(TransomOffset)) AND NOT(EXISTS(TransomThickness)));
WR34 : (EXISTS(CasingDepth) AND EXISTS(CasingThickness)) XOR
        (NOT(EXISTS(CasingDepth)) AND NOT(EXISTS(CasingThickness)));
WR35 : (EXISTS(SELFWIfcPropertySetDefinition.DefinesType[1]))
AND
(
('IFC4.IFCD00RTYPE' IN TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1]))
OR
('IFC4.IFCD00RSTYLE' IN TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1]))
);
END_ENTITY;

```

ENTITY IfcDoorPanelProperties

SUBTYPE OF(IfcPreDefinedPropertySet);

PanelDepth : OPTIONAL IfcPositiveLengthMeasure;

PanelOperation : IfcDoorPanelOperationEnum;

PanelWidth : OPTIONAL IfcNormalisedRatioMeasure;

PanelPosition : IfcDoorPanelPositionEnum;

ShapeAspectStyle : OPTIONAL IfcShapeAspect;

WHERE

APPLICABLETOTYPE : (EXISTS(SELFWIfcPropertySetDefinition.DefinesType[1]))

AND

(

TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1])) IN

OR

TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1])) IN

);

END_ENTITY;

ENTITY IfcPermeableCoveringProperties

SUBTYPE OF(IfcPreDefinedPropertySet);

OperationType : IfcPermeableCoveringOperationEnum;

PanelPosition : IfcWindowPanelPositionEnum;

FrameDepth : OPTIONAL IfcPositiveLengthMeasure;

FrameThickness : OPTIONAL IfcPositiveLengthMeasure;

ShapeAspectStyle : OPTIONAL IfcShapeAspect;

END_ENTITY;

ENTITY IfcReinforcementDefinitionProperties

SUBTYPE OF(IfcPreDefinedPropertySet);

DefinitionType : OPTIONAL IfcLabel;

ReinforcementSectionDefinitions : LIST [1:?] OF IfcSectionReinforcementProperties;

END_ENTITY;

ENTITY IfcWindowLiningProperties

SUBTYPE OF(IfcPreDefinedPropertySet);

LiningDepth : OPTIONAL IfcPositiveLengthMeasure;

LiningThickness : OPTIONAL IfcNonNegativeLengthMeasure;

TransomThickness : OPTIONAL IfcNonNegativeLengthMeasure;

MullionThickness : OPTIONAL IfcNonNegativeLengthMeasure;

FirstTransomOffset : OPTIONAL IfcNormalisedRatioMeasure;

SecondTransomOffset : OPTIONAL IfcNormalisedRatioMeasure;

FirstMullionOffset : OPTIONAL IfcNormalisedRatioMeasure;

SecondMullionOffset : OPTIONAL IfcNormalisedRatioMeasure;

ShapeAspectStyle : OPTIONAL IfcShapeAspect;

LiningOffset : OPTIONAL IfcLengthMeasure;

LiningToPanelOffsetX : OPTIONAL IfcLengthMeasure;

LiningToPanelOffsetY : OPTIONAL IfcLengthMeasure;

WHERE

WR31 : NOT(EXISTS(LiningDepth) AND NOT(EXISTS(LiningThickness)));

WR32 : NOT(NOT(EXISTS(FirstTransomOffset)) AND EXISTS(SecondTransomOffset));

WR33 : NOT(NOT(EXISTS(FirstMullionOffset)) AND EXISTS(SecondMullionOffset));

WR34 : (EXISTS(SELFWIfcPropertySetDefinition.DefinesType[1]))

AND

(

('IFC4.IFCWINDOWTYPE' IN TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1]))

OR

('IFC4.IFCWINDOWSTYLE' IN TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1]))

);

END_ENTITY;

ENTITY IfcWindowPanelProperties

SUBTYPE OF(IfcPreDefinedPropertySet);

OperationType : IfcWindowPanelOperationEnum;

PanelPosition : IfcWindowPanelPositionEnum;

FrameDepth : OPTIONAL IfcPositiveLengthMeasure;

FrameThickness : OPTIONAL IfcPositiveLengthMeasure;

ShapeAspectStyle : OPTIONAL IfcShapeAspect;

WHERE

APPLICABLETOTYPE : (EXISTS(SELFWIfcPropertySetDefinition.DefinesType[1]))

AND

(

TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1])) IN

OR

TYPEOF(SELFWIfcPropertySetDefinition.DefinesType[1])) IN

);

END_ENTITY;

ENTITY IfcQuantitySet

ABSTRACT SUPERTYPE

SUBTYPE OF(IfcPropertySetDefinition);

END_ENTITY;

ENTITY IfcElementQuantity

SUBTYPE OF(IfcQuantitySet);

MethodOfMeasurement : OPTIONAL IfcLabel;

Quantities : SET [1:?] OF IfcPhysicalQuantity;

WHERE

UNIQUEQUANTITYNAMES : IfcUniqueQuantityNames(Quantities);

END_ENTITY;

ENTITY IfcMaterialClassificationRelationship;

MaterialClassifications : SET [1:?] OF IfcClassificationSelect;

ClassifiedMaterial : IfcMaterial;

END_ENTITY;

RULE IfcRepresentationContextSameWCS FOR

(IfcGeometricRepresentationContext);

LOCAL

IsDifferent : LOGICAL := FALSE;

END_LOCAL;

IF (SIZEOF(IfcGeometricRepresentationContext) > 1)

THEN

REPEAT i := 2 TO HI INDEX(IfcGeometricRepresentationContext);

IF (IfcGeometricRepresentationContext[1].WorldCoordinateSystem
IfcGeometricRepresentationContext[i].WorldCoordinateSystem) :<>

```

        THEN
            IsDifferent                                     :=
(NOT(IfcSameValidPrecision(IfcGeometricRepresentationContext[1].Precision,
IfcGeometricRepresentationContext[i].Precision)))

            OR
(NOT(IfcSameAxis2Placement(IfcGeometricRepresentationContext[1].WorldCoordinateSystem,
IfcGeometricRepresentationContext[i].WorldCoordinateSystem,
IfcGeometricRepresentationContext[1].Precision)));

        IF (IsDifferent = TRUE) THEN
            ESCAPE;
        END_IF;
    END_IF;
END_REPEAT;
END_IF;
WHERE
    WR1 : IsDifferent = FALSE;
END_RULE;

RULE IfcSingleProjectInstance FOR
    (IfcProject);
WHERE
    WR1 : SIZEOF(IfcProject) <= 1;
END_RULE;

FUNCTION IfcBaseAxis
(Dim : INTEGER;
 Axis1, Axis2, Axis3 : IfcDirection)
: LIST [2:3] OF IfcDirection;

```

LOCAL

U : LIST [2:3] OF IfcDirection;

Factor : REAL;

D1, D2 : IfcDirection;

END_LOCAL;

IF (Dim = 3) THEN

D1 := NVL(IfcNormalise(Axis3), IfcRepresentationItem() || IfcGeometricRepresentationItem ()
|| IfcDirection([0.0,0.0,1.0]));

D2 := IfcFirstProjAxis(D1, Axis1);

U := [D2, IfcSecondProjAxis(D1, D2, Axis2), D1];

ELSE

IF EXISTS(Axis1) THEN

D1 := IfcNormalise(Axis1);

U := [D1, IfcOrthogonalComplement(D1)];

IF EXISTS(Axis2) THEN

Factor := IfcDotProduct(Axis2, U[2]);

IF (Factor < 0.0) THEN

U[2].DirectionRatios[1] := -U[2].DirectionRatios[1];

U[2].DirectionRatios[2] := -U[2].DirectionRatios[2];

END_IF;

END_IF;

ELSE

IF EXISTS(Axis2) THEN

D1 := IfcNormalise(Axis2);

U := [IfcOrthogonalComplement(D1), D1];

U[1].DirectionRatios[1] := -U[1].DirectionRatios[1];

U[1].DirectionRatios[2] := -U[1].DirectionRatios[2];

ELSE

```

        U := [IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([1.0, 0.0]),
            IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([0.0, 1.0])];
    END_IF;
    END_IF;
    END_IF;
    RETURN(U);
END_FUNCTION;

```

```

FUNCTION IfcBooleanChoose

```

```

(B : BOOLEAN ;
    Choice1, Choice2 : Generic : Item) : Generic : Item;
    IF B THEN
        RETURN (Choice1);
    ELSE
        RETURN (Choice2);
    END_IF;
END_FUNCTION;

```

```

FUNCTION IfcBuild2Axes

```

```

(RefDirection : IfcDirection)
    : LIST [2:2] OF IfcDirection;
    LOCAL
        D : IfcDirection := NVL(IfcNormalise(RefDirection),
            IfcRepresentationItem() || IfcGeometricRepresentationItem () || IfcDirection([1.0,0.0]));
    END_LOCAL;
    RETURN([D, IfcOrthogonalComplement(D)]);
END_FUNCTION;

```

```

FUNCTION IfcBuildAxes
(Axis, RefDirection : IfcDirection)
  : LIST [3:3] OF IfcDirection;
LOCAL
  D1, D2 : IfcDirection;
END_LOCAL;
  D1 := NVL(IfcNormalise(Axis), IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([0.0,0.0,1.0]));
  D2 := IfcFirstProjAxis(D1, RefDirection);
  RETURN ([D2, IfcNormalise(IfcCrossProduct(D1,D2))WIfcVector.Orientation, D1]);
END_FUNCTION;

FUNCTION IfcConsecutiveSegments
(Segments : LIST [1:?] OF IfcSegmentIndexSelect)
  : BOOLEAN;
LOCAL
  Result : BOOLEAN := TRUE;
END_LOCAL;

REPEAT i := 1 to (HIINDEX(Segments)-1);
  IF Segments[i][HIINDEX(Segments[i])] <> Segments[i+1][1] THEN
    BEGIN
      Result := FALSE;
      ESCAPE;
    END;
  END_IF;
END_REPEAT;

RETURN (Result);

```

```
END_FUNCTION;
```

```
FUNCTION IfcConstraintsParamBSpline
```

```
( Degree, UpKnots, UpCp : INTEGER;
```

```
  KnotMult : LIST OF INTEGER;
```

```
  Knots : LIST OF IfcParameterValue )
```

```
: BOOLEAN;
```

```
LOCAL
```

```
  Result : BOOLEAN := TRUE;
```

```
  K, Sum : INTEGER;
```

```
END_LOCAL;
```

```
(* Find sum of knot multiplicities. *)
```

```
Sum := KnotMult[1];
```

```
REPEAT i := 2 TO UpKnots;
```

```
  Sum := Sum + KnotMult[i];
```

```
END_REPEAT;
```

```
(* Check limits holding for all B-spline parametrisations *)
```

```
IF (Degree < 1) OR (UpKnots < 2) OR (UpCp < Degree) OR
```

```
  (Sum <> (Degree + UpCp + 2)) THEN
```

```
  Result := FALSE;
```

```
  RETURN(Result);
```

```
END_IF;
```

```
K := KnotMult[1];
```

```
IF (K < 1) OR (K > Degree + 1) THEN
```

```
  Result := FALSE;
```

```

RETURN(Result);
END_IF;

REPEAT i := 2 TO UpKnots;
  IF (KnotMult[i] < 1) OR (Knots[i] <= Knots[i-1]) THEN
    Result := FALSE;
    RETURN(Result);
  END_IF;
  K := KnotMult[i];
  IF (i < UpKnots) AND (K > Degree) THEN
    Result := FALSE;
    RETURN(Result);
  END_IF;
  IF (i = UpKnots) AND (K > Degree + 1) THEN
    Result := FALSE;
    RETURN(Result);
  END_IF;
END_REPEAT;

RETURN(result);
END_FUNCTION;

```

```

FUNCTION IfcConvertDirectionInto2D
(Direction : IfcDirection)
: IfcDirection;

LOCAL
  Direction2D : IfcDirection := IfcRepresentationItem() || IfcGeometricRepresentationItem ()
|| IfcDirection([0.,1.]);
END_LOCAL;

```



```

Direction2D.DirectionRatios[1] := Direction.DirectionRatios[1];
Direction2D.DirectionRatios[2] := Direction.DirectionRatios[2];

RETURN (Direction2D);
END_FUNCTION;

```

```

FUNCTION IfcCorrectDimensions

```

```

(m : IfcUnitEnum; Dim : IfcDimensionalExponents) : LOGICAL;

```

```

CASE m OF

```

```

    LENGTHUNIT : IF

```

```

        Dim = (IfcDimensionalExponents (1, 0, 0, 0, 0, 0, 0))

```

```

        THEN RETURN(TRUE);

```

```

        ELSE RETURN(FALSE);

```

```

    END_IF;

```

```

    MASSUNIT : IF

```

```

        Dim = (IfcDimensionalExponents (0, 1, 0, 0, 0, 0, 0))

```

```

        THEN RETURN(TRUE);

```

```

        ELSE RETURN(FALSE);

```

```

    END_IF;

```

```

    TIMEUNIT : IF

```

```

        Dim = (IfcDimensionalExponents (0, 0, 1, 0, 0, 0, 0))

```

```

        THEN RETURN(TRUE);

```

```

        ELSE RETURN(FALSE);

```

```

    END_IF;

```

```

    ELECTRICCURRENTUNIT : IF

```

```

        Dim = (IfcDimensionalExponents (0, 0, 0, 1, 0, 0, 0))

```

```

        THEN RETURN(TRUE);

```

```

        ELSE RETURN(FALSE);

```

```

    END_IF;

```

THERMODYNAMICTEMPERATUREUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, 0, 0, 1, 0, 0))  
  THEN RETURN(TRUE);  
  ELSE RETURN(FALSE);  
END_IF;
```

AMOUNTOFSUBSTANCEUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, 0, 0, 0, 1, 0))  
  THEN RETURN(TRUE);  
  ELSE RETURN(FALSE);  
END_IF;
```

LUMINOUSINTENSITYUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, 0, 0, 0, 0, 1))  
  THEN RETURN(TRUE);  
  ELSE RETURN(FALSE);  
END_IF;
```

PLANEANGLEUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, 0, 0, 0, 0, 0))  
  THEN RETURN(TRUE);  
  ELSE RETURN(FALSE);  
END_IF;
```

SOLIDANGLEUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, 0, 0, 0, 0, 0))  
  THEN RETURN(TRUE);  
  ELSE RETURN(FALSE);  
END_IF;
```

AREAUNIT : IF

```
Dim = (IfcDimensionalExponents (2, 0, 0, 0, 0, 0, 0))  
  THEN RETURN(TRUE);  
  ELSE RETURN(FALSE);  
END_IF;
```

VOLUMEUNIT : IF

```
Dim = (IfcDimensionalExponents (3, 0, 0, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
```

ABSORBEDDOSEUNIT : IF

```
Dim = (IfcDimensionalExponents (2, 0, -2, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
```

RADIOACTIVITYUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, -1, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
```

ELECTRICCAPACITANCEUNIT : IF

```
Dim = (IfcDimensionalExponents (-2, -1, 4, 2, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
```

DOSEEQUIVALENTUNIT : IF

```
Dim = (IfcDimensionalExponents (2, 0, -2, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
```

ELECTRICCHARGEUNIT : IF

```
Dim = (IfcDimensionalExponents (0, 0, 1, 1, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
```

```

END_IF;
ELECTRICCONDUCTANCEUNIT : IF
  Dim = (IfcDimensionalExponents (-2, -1, 3, 2, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
ELECTRICVOLTAGEUNIT : IF
  Dim = (IfcDimensionalExponents (2, 1, -3, -1, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
ELECTRICRESISTANCEUNIT : IF
  Dim = (IfcDimensionalExponents (2, 1, -3, -2, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
ENERGYUNIT : IF
  Dim = (IfcDimensionalExponents (2, 1, -2, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
FORCEUNIT : IF
  Dim = (IfcDimensionalExponents (1, 1, -2, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
FREQUENCYUNIT : IF
  Dim = (IfcDimensionalExponents (0, 0, -1, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);

```

```

END_IF;
INDUCTANCEUNIT : IF
  Dim = (IfcDimensionalExponents (2, 1, -2, -2, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
ILLUMINANCEUNIT : IF
  Dim = (IfcDimensionalExponents (-2, 0, 0, 0, 0, 0, 1))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
LUMINOUSFLUXUNIT : IF
  Dim = (IfcDimensionalExponents (0, 0, 0, 0, 0, 0, 1))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
MAGNETICFLUXUNIT : IF
  Dim = (IfcDimensionalExponents (2, 1, -2, -1, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
MAGNETICFLUXDENSITYUNIT : IF
  Dim = (IfcDimensionalExponents (0, 1, -2, -1, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);
END_IF;
POWERUNIT : IF
  Dim = (IfcDimensionalExponents (2, 1, -3, 0, 0, 0, 0))
  THEN RETURN(TRUE);
  ELSE RETURN(FALSE);

```

```

    END_IF;
PRESSUREUNIT : IF
    Dim = (IfcDimensionalExponents (-1, 1, -2, 0, 0, 0, 0))
    THEN RETURN(TRUE);
    ELSE RETURN(FALSE);
    END_IF;

    OTHERWISE :
        RETURN (UNKNOWN);
END_CASE;
END_FUNCTION;

FUNCTION IfcCorrectFillAreaStyle
(Styles : SET[1:?] OF IfcFillStyleSelect)
:LOGICAL;

LOCAL
    Hatching : INTEGER := 0;
    Tiles     : INTEGER := 0;
    Colour    : INTEGER := 0;
    External  : INTEGER := 0;
END_LOCAL;

External := SIZEOF(QUERY(Style <* Styles |
    'IFC4.IFCEXTERNALLYDEFINEDHATCHSTYLE' IN
    TYPEOF(Style)));

Hatching := SIZEOF(QUERY(Style <* Styles |
    'IFC4.IFCFILLAREASTYLEHATCHING' IN

```

```

    TYPEOF(Style));

Tiles    := SIZEOF(QUERY(Style <* Styles |
    'IFC4.IFCFILLAREASTYLETILES' IN
    TYPEOF(Style)));

Colour   := SIZEOF(QUERY(Style <* Styles |
    'IFC4.IFCCOLOUR' IN
    TYPEOF(Style)));

IF (External > 1) THEN
    RETURN (FALSE);
END_IF;

IF ((External = 1) AND ((Hatching > 0) OR (Tiles > 0) OR (Colour > 0))) THEN
    RETURN (FALSE);
END_IF;

IF (Colour > 1) THEN
    RETURN (FALSE);
END_IF;

IF ((Hatching > 0) AND (Tiles >0)) THEN
    RETURN (FALSE);
END_IF;

RETURN(TRUE);

```

END_FUNCTION;

FUNCTION IfcCorrectLocalPlacement

(AxisPlacement: IfcAxis2Placement;

RelPlacement : IfcObjectPlacement):LOGICAL;

IF (EXISTS(RelPlacement)) THEN

IF ('IFC4.IFCGRIDPLACEMENT' IN TYPEOF(RelPlacement)) THEN

RETURN(?);

END_IF;

IF ('IFC4.IFCLOCALPLACEMENT' IN TYPEOF(RelPlacement)) THEN

IF ('IFC4.IFCAXIS2PLACEMENT2D' IN TYPEOF(AxisPlacement)) THEN

RETURN(TRUE);

END_IF;

IF ('IFC4.IFCAXIS2PLACEMENT3D' IN TYPEOF(AxisPlacement)) THEN

IF (RelPlacement.IfcLocalPlacement.RelativePlacement.Dim = 3) THEN

RETURN(TRUE);

ELSE

RETURN(FALSE);

END_IF;

END_IF;

END_IF;

ELSE

RETURN(TRUE);

END_IF;

RETURN(?);

END_FUNCTION;

FUNCTION IfcCorrectObjectAssignment

(Constraint: IfcObjectTypeEnum; Objects : SET[1:?] OF IfcObjectDefinition)


```
: LOGICAL ;
```

```
LOCAL
```

```
Count : INTEGER := 0;
```

```
END_LOCAL;
```

```
IF NOT(EXISTS(Constraint)) THEN
```

```
RETURN(TRUE);
```

```
END_IF;
```

```
CASE Constraint OF
```

```
  IfcObjectTypeEnum.NOTDEFINED : RETURN(TRUE);
```

```
  IfcObjectTypeEnum.PRODUCT :
```

```
    BEGIN
```

```
      Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCPRODUCT' IN TYPEOF(temp))));
```

```
      RETURN(Count = 0);
```

```
    END;
```

```
  IfcObjectTypeEnum.PROCESS :
```

```
    BEGIN
```

```
      Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCPROCESS' IN TYPEOF(temp))));
```

```
      RETURN(Count = 0);
```

```
    END;
```

```
  IfcObjectTypeEnum.CONTROL :
```

```
    BEGIN
```

```
      Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCCONTROL' IN TYPEOF(temp))));
```

```
      RETURN(Count = 0);
```

```
    END;
```

```
  IfcObjectTypeEnum.RESOURCE :
```

```
    BEGIN
```

```
      Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCRESOURCE' IN TYPEOF(temp))));
```

```

        RETURN(Count = 0);
    END;
IfcObjectTypeEnum.ACTOR :
    BEGIN
        Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCACTOR' IN TYPEOF(temp))));
        RETURN(Count = 0);
    END;
IfcObjectTypeEnum.GROUP :
    BEGIN
        Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCGROUP' IN TYPEOF(temp))));
        RETURN(Count = 0);
    END;
IfcObjectTypeEnum.PROJECT :
    BEGIN
        Count := SIZEOF(QUERY(temp <* Objects | NOT('IFC4.IFCPROJECT' IN TYPEOF(temp))));
        RETURN(Count = 0);
    END;
    OTHERWISE : RETURN(?);
END_CASE;
END_FUNCTION;

```

```

FUNCTION IfcCorrectUnitAssignment

```

```

(Units : SET [1:?] OF IfcUnit)

```

```

: LOGICAL;

```

```

LOCAL

```

```

    NamedUnitNumber : INTEGER := 0;

```

```

    DerivedUnitNumber : INTEGER := 0;

```

```

    MonetaryUnitNumber : INTEGER := 0;

```

```

    NamedUnitNames : SET OF IfcUnitEnum := [];

```

```

    DerivedUnitNames    : SET OF IfcDerivedUnitEnum := [];
END_LOCAL;

NamedUnitNumber      := SIZEOF(QUERY(temp <* Units | ('IFC4.IFCNAMEDUNIT' IN TYPEOF(temp)) AND
NOT(temp\IfcNamedUnit.UnitType = IfcUnitEnum.USERDEFINED)));

DerivedUnitNumber    := SIZEOF(QUERY(temp <* Units | ('IFC4.IFCDERIVEDUNIT' IN TYPEOF(temp)) AND
NOT(temp\IfcDerivedUnit.UnitType = IfcDerivedUnitEnum.USERDEFINED)));

MonetaryUnitNumber  := SIZEOF(QUERY(temp <* Units | 'IFC4.IFCMONETARYUNIT' IN TYPEOF(temp)));

REPEAT i := 1 TO SIZEOF(Units);
    IF (('IFC4.IFCNAMEDUNIT' IN TYPEOF(Units[i])) AND NOT(Units[i]\IfcNamedUnit.UnitType =
IfcUnitEnum.USERDEFINED)) THEN
        NamedUnitNames := NamedUnitNames + Units[i]\IfcNamedUnit.UnitType;
    END_IF;

    IF (('IFC4.IFCDERIVEDUNIT' IN TYPEOF(Units[i])) AND NOT(Units[i]\IfcDerivedUnit.UnitType =
IfcDerivedUnitEnum.USERDEFINED)) THEN
        DerivedUnitNames := DerivedUnitNames + Units[i]\IfcDerivedUnit.UnitType;
    END_IF;
END_REPEAT;

RETURN((SIZEOF(NamedUnitNames) = NamedUnitNumber) AND (SIZEOF(DerivedUnitNames) =
DerivedUnitNumber) AND (MonetaryUnitNumber <= 1));

END_FUNCTION;

FUNCTION IfcCrossProduct
(Arg1, Arg2 : IfcDirection)
: IfcVector;
LOCAL
Mag : REAL;
Res : IfcDirection;
V1,V2 : LIST[3:3] OF REAL;

```

```

Result : IfcVector;
END_LOCAL;

IF (NOT EXISTS (Arg1) OR (Arg1.Dim = 2)) OR (NOT EXISTS (Arg2) OR (Arg2.Dim = 2)) THEN
    RETURN(?);
ELSE
    BEGIN
        V1 := IfcNormalise(Arg1)WIfcDirection.DirectionRatios;

        V2 := IfcNormalise(Arg2)WIfcDirection.DirectionRatios;
        Res := IfcRepresentationItem() || IfcGeometricRepresentationItem (
            || IfcDirection([(V1[2]*V2[3] - V1[3]*V2[2]), (V1[3]*V2[1] - V1[1]*V2[3]),
            (V1[1]*V2[2] - V1[2]*V2[1])]);
        Mag := 0.0;
        REPEAT i := 1 TO 3;
            Mag := Mag + Res.DirectionRatios[i]*Res.DirectionRatios[i];
        END_REPEAT;
        IF (Mag > 0.0) THEN
            Result := IfcRepresentationItem() || IfcGeometricRepresentationItem ( || IfcVector(Res,
            SQRT(Mag));
        ELSE
            Result := IfcRepresentationItem() || IfcGeometricRepresentationItem ( || IfcVector(Arg1,
            0.0);
        END_IF;
        RETURN(Result);
    END;
END_IF;
END_FUNCTION;

FUNCTION IfcCurveDim
(Curve : IfcCurve)

```

```

        : IfcDimensionCount;

IF ('IFC4.IFCLINE' IN TYPEOF(Curve))
    THEN RETURN(CurveWIfcLine.Pnt.Dim);
END_IF;
IF ('IFC4.IFCCONIC' IN TYPEOF(Curve))
    THEN RETURN(CurveWIfcConic.Position.Dim);
END_IF;
IF ('IFC4.IFCPOLYLINE' IN TYPEOF(Curve))
    THEN RETURN(CurveWIfcPolyline.Points[1].Dim);
END_IF;
IF ('IFC4.IFCTRIMMEDCURVE' IN TYPEOF(Curve))
    THEN RETURN(IfcCurveDim(CurveWIfcTrimmedCurve.BasisCurve));
END_IF;
IF ('IFC4.IFCCOMPOSITECURVE' IN TYPEOF(Curve))
    THEN RETURN(CurveWIfcCompositeCurve.Segments[1].Dim);
END_IF;
IF ('IFC4.IFCBSPLINECURVE' IN TYPEOF(Curve))
    THEN RETURN(CurveWIfcBSplineCurve.ControlPointsList[1].Dim);
END_IF;
IF ('IFC4.IFCOFFSETCURVE2D' IN TYPEOF(Curve))
    THEN RETURN(2);
END_IF;
IF ('IFC4.IFCOFFSETCURVE3D' IN TYPEOF(Curve))
    THEN RETURN(3);
END_IF;
IF ('IFC4.IFCPCURVE' IN TYPEOF(Curve))
    THEN RETURN(3);
END_IF;
IF ('IFC4.IFCINDEXEDPOLYCURVE' IN TYPEOF(Curve))

```

```

        THEN RETURN(CurveWeightsIndexedPolyCurve.Points.Dim);
    END_IF;
RETURN (?);
END_FUNCTION;

FUNCTION IfcCurveWeightsPositive
( B: IfcRationalBSplineCurveWithKnots)
: BOOLEAN;

    LOCAL
        Result : BOOLEAN := TRUE;
    END_LOCAL;

    REPEAT i := 0 TO B.UpperIndexOnControlPoints;
        IF B.Weights[i] <= 0.0 THEN
            Result := FALSE;
            RETURN(Result);
        END_IF;
    END_REPEAT;
    RETURN(Result);
END_FUNCTION;

FUNCTION IfcDeriveDimensionalExponents
(UnitElements : SET [1:?] OF IfcDerivedUnitElement)
: IfcDimensionalExponents;

    LOCAL
        Result : IfcDimensionalExponents :=
            IfcDimensionalExponents(0, 0, 0, 0, 0, 0, 0);
    END_LOCAL;
    REPEAT i := LOINDEX(UnitElements) TO HIINDEX(UnitElements);

```

```

Result.LengthExponent := Result.LengthExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.LengthExponent);
Result.MassExponent := Result.MassExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.MassExponent);
Result.TimeExponent := Result.TimeExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.TimeExponent);
Result.ElectricCurrentExponent := Result.ElectricCurrentExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.ElectricCurrentExponent);
Result.ThermodynamicTemperatureExponent := Result.ThermodynamicTemperatureExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.ThermodynamicTemperatureExponent);
Result.AmountOfSubstanceExponent := Result.AmountOfSubstanceExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.AmountOfSubstanceExponent);
Result.LuminousIntensityExponent := Result.LuminousIntensityExponent +
    (UnitElements[i].Exponent *
        UnitElements[i].Unit.Dimensions.LuminousIntensityExponent);
END_REPEAT;
RETURN (Result);
END_FUNCTION;

FUNCTION IfcDimensionsForSiUnit
(n : IfcSiUnitName ) : IfcDimensionalExponents;
CASE n OF
    METRE : RETURN (IfcDimensionalExponents
        (1, 0, 0, 0, 0, 0, 0));

```

SQUARE_METRE : RETURN (IfcDimensionalExponents
 (2, 0, 0, 0, 0, 0, 0));

CUBIC_METRE : RETURN (IfcDimensionalExponents
 (3, 0, 0, 0, 0, 0, 0));

GRAM : RETURN (IfcDimensionalExponents
 (0, 1, 0, 0, 0, 0, 0));

SECOND : RETURN (IfcDimensionalExponents
 (0, 0, 1, 0, 0, 0, 0));

AMPERE : RETURN (IfcDimensionalExponents
 (0, 0, 0, 1, 0, 0, 0));

KELVIN : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 1, 0, 0));

MOLE : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 0, 1, 0));

CANDELA : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 0, 0, 1));

RADIAN : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 0, 0, 0));

STERADIAN : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 0, 0, 0));

HERTZ : RETURN (IfcDimensionalExponents
 (0, 0, -1, 0, 0, 0, 0));

NEWTON : RETURN (IfcDimensionalExponents
 (1, 1, -2, 0, 0, 0, 0));

PASCAL : RETURN (IfcDimensionalExponents
 (-1, 1, -2, 0, 0, 0, 0));

JOULE : RETURN (IfcDimensionalExponents
 (2, 1, -2, 0, 0, 0, 0));

WATT : RETURN (IfcDimensionalExponents
 (2, 1, -3, 0, 0, 0, 0));

COULOMB : RETURN (IfcDimensionalExponents
 (0, 0, 1, 1, 0, 0, 0));

VOLT : RETURN (IfcDimensionalExponents
 (2, 1, -3, -1, 0, 0, 0));

FARAD : RETURN (IfcDimensionalExponents
 (-2, -1, 4, 2, 0, 0, 0));

OHM : RETURN (IfcDimensionalExponents
 (2, 1, -3, -2, 0, 0, 0));

SIEMENS : RETURN (IfcDimensionalExponents
 (-2, -1, 3, 2, 0, 0, 0));

WEBER : RETURN (IfcDimensionalExponents
 (2, 1, -2, -1, 0, 0, 0));

TESLA : RETURN (IfcDimensionalExponents
 (0, 1, -2, -1, 0, 0, 0));

HENRY : RETURN (IfcDimensionalExponents
 (2, 1, -2, -2, 0, 0, 0));

DEGREE_CELSIUS : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 1, 0, 0));

LUMEN : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 0, 0, 1));

LUX : RETURN (IfcDimensionalExponents
 (-2, 0, 0, 0, 0, 0, 1));

BECQUEREL : RETURN (IfcDimensionalExponents
 (0, 0, -1, 0, 0, 0, 0));

GRAY : RETURN (IfcDimensionalExponents
 (2, 0, -2, 0, 0, 0, 0));

SIEVERT : RETURN (IfcDimensionalExponents
 (2, 0, -2, 0, 0, 0, 0));

OTHERWISE : RETURN (IfcDimensionalExponents
 (0, 0, 0, 0, 0, 0, 0));

```

END_CASE;
END_FUNCTION;

FUNCTION IfcDotProduct
(Arg1, Arg2 : IfcDirection)
  : REAL;
LOCAL
  Scalar : REAL;
  Vec1, Vec2 : IfcDirection;
  Ndim : INTEGER;
END_LOCAL;

IF NOT EXISTS (Arg1) OR NOT EXISTS (Arg2) THEN
  Scalar := ?;
ELSE
  IF (Arg1.Dim <> Arg2.Dim) THEN
    Scalar := ?;
  ELSE
    BEGIN
      Vec1 := IfcNormalise(Arg1);
      Vec2 := IfcNormalise(Arg2);
      Ndim := Arg1.Dim;
      Scalar := 0.0;
      REPEAT i := 1 TO Ndim;
        Scalar := Scalar + Vec1.DirectionRatios[i]*Vec2.DirectionRatios[i];
      END_REPEAT;
    END;
  END_IF;
END_IF;
RETURN (Scalar);

```

END_FUNCTION;

FUNCTION IfcFirstProjAxis

(ZAxis, Arg : IfcDirection) : IfcDirection;

LOCAL

XAxis : IfcDirection;

V : IfcDirection;

Z : IfcDirection;

XVec : IfcVector;

END_LOCAL;

IF (NOT EXISTS(ZAxis)) THEN

RETURN (?);

ELSE

Z := IfcNormalise(ZAxis);

IF NOT EXISTS(Arg) THEN

IF (Z.DirectionRatios \neq [1.0,0.0,0.0]) THEN

V := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([1.0,0.0,0.0]);

ELSE

V := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([0.0,1.0,0.0]);

END_IF;

ELSE

IF (Arg.Dim \neq 3) THEN

RETURN (?);

END_IF;

IF ((IfcCrossProduct(Arg,Z).Magnitude) = 0.0) THEN

RETURN (?);

ELSE

```

        V := IfcNormalise(Arg);
    END_IF;
END_IF;
XVec := IfcScalarTimesVector(IfcDotProduct(V, Z), Z);
XAxis := IfcVectorDifference(V, XVec).Orientation;
XAxis := IfcNormalise(XAxis);
END_IF;
RETURN(XAxis);
END_FUNCTION;

FUNCTION IfcGetBasisSurface
(C : IfcCurveOnSurface) : SET[0:2] OF IfcSurface;

LOCAL
    Surfs : SET[0:2] OF IfcSurface;
    N : INTEGER;
END_LOCAL;

Surfs := [];
IF 'IFC4.IFCPCURVE' IN TYPEOF (C) THEN
    Surfs := [CWIfcPCurve.BasisSurface];
ELSE
    IF 'IFC4.IFCCOMPOSITECURVEONSURFACE' IN TYPEOF (C) THEN

        (* For an IfcCompositeCurveOnSurface the BasisSurface is the intersection
           of the BasisSurface of all the segments. *)

        N := SIZEOF(CWIfcCompositeCurve.Segments);
        Surfs := IfcGetBasisSurface(CWIfcCompositeCurve.Segments[1].ParentCurve);
    END_IF;
END_IF;

```

```

IF N > 1 THEN
  REPEAT i := 2 TO N;
    Surfs := Surfs * IfcGetBasisSurface(CWIfcCompositeCurve.Segments[1].ParentCurve);
  END_REPEAT;
END_IF;
END_IF;
END_IF;
RETURN(Surfs);
END_FUNCTION;

```

```

FUNCTION IfcListToArray
(Lis : LIST [0:?] OF GENERIC : T;
  Low,U : INTEGER) : ARRAY OF GENERIC : T;
LOCAL
  N : INTEGER;
  Res : ARRAY [Low:U] OF GENERIC : T;
END_LOCAL;

N := SIZEOF(Lis);
IF (N <> (U-Low +1)) THEN
  RETURN(?);
ELSE
  Res := [Lis[1] : N];
  REPEAT i := 2 TO N;
    Res[Low+i-1] := Lis[i];
  END_REPEAT;
  RETURN(Res);
END_IF;
END_FUNCTION;

```

```

FUNCTION IfcLoopHeadToTail
(ALoop : IfcEdgeLoop) : LOGICAL;
  LOCAL
    N : INTEGER;
    P : LOGICAL := TRUE;
  END_LOCAL;

  N := SIZEOF (ALoop.EdgeList);
  REPEAT i := 2 TO N;
    P := P AND (ALoop.EdgeList[i-1].EdgeEnd :=:
                ALoop.EdgeList[i].EdgeStart);
  END_REPEAT;
  RETURN (P);
END_FUNCTION;

```

```

FUNCTION IfcMakeArrayOfArray
(Lis : LIST[1:?] OF LIST [1:?] OF GENERIC : T;
 Low1, U1, Low2, U2 : INTEGER):
ARRAY [Low1:U1] OF ARRAY [Low2:U2] OF GENERIC : T;

  LOCAL
    Res : ARRAY[Low1:U1] OF ARRAY [Low2:U2] OF GENERIC : T;
  END_LOCAL;

  (* Check input dimensions for consistency *)
  IF (U1-Low1+1) <> SIZEOF(Lis) THEN
    RETURN (?);
  END_IF;
  IF (U2 - Low2 + 1 ) <> SIZEOF(Lis[1]) THEN
    RETURN (?) ;

```

```

END_IF;

(* Initialise Res with values from Lis[1] *)
Res := [IfcListToArray(Lis[1], Low2, U2) : (U1-Low1 + 1)];
REPEAT i := 2 TO HIINDEX(Lis);
  IF (U2-Low2+1) <> SIZEOF(Lis[i]) THEN
    RETURN (?);
  END_IF;
  Res[Low1+i-1] := IfcListToArray(Lis[i], Low2, U2);
END_REPEAT;
RETURN (Res);
END_FUNCTION;

FUNCTION IfcMIsTotalThickness
(LayerSet : IfcMaterialLayerSet) : IfcLengthMeasure;
LOCAL
  Max : IfcLengthMeasure := LayerSet.MaterialLayers[1].LayerThickness;
END_LOCAL;

IF SIZEOF(LayerSet.MaterialLayers) > 1 THEN
  REPEAT i := 2 TO HIINDEX(LayerSet.MaterialLayers);
    Max := Max + LayerSet.MaterialLayers[i].LayerThickness;
  END_REPEAT;
END_IF;
RETURN (Max);
END_FUNCTION;

FUNCTION IfcNormalise
(Arg : IfcVectorOrDirection)
: IfcVectorOrDirection;

```

```

LOCAL
  Ndim : INTEGER;
  V    : IfcDirection
        := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([1.,0.]);
  Vec  : IfcVector
        := IfcRepresentationItem() || IfcGeometricRepresentationItem () || IfcVector (
          IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([1.,0.]), 1.);
  Mag  : REAL;
  Result : IfcVectorOrDirection
        := V;
END_LOCAL;

IF NOT EXISTS (Arg) THEN
  RETURN (?);
ELSE
  IF 'IFC4.IFCVECTOR' IN TYPEOF(Arg) THEN
    BEGIN
      Ndim := ArgWIfcVector.Dim;
      V.DirectionRatios := ArgWIfcVector.Orientation.DirectionRatios;
      Vec.Magnitude := ArgWIfcVector.Magnitude;
      Vec.Orientation := V;
      IF ArgWIfcVector.Magnitude = 0.0 THEN
        RETURN(?);
      ELSE
        Vec.Magnitude := 1.0;
      END_IF;
    END;
  ELSE
    RETURN(?);
  END;
ELSE
  RETURN(?);
END;

```



```

BEGIN
    Ndim := ArgWlfcDirection.Dim;
    V.DirectionRatios := ArgWlfcDirection.DirectionRatios;
END;
END_IF;

Mag := 0.0;
REPEAT i := 1 TO Ndim;
    Mag := Mag + V.DirectionRatios[i]*V.DirectionRatios[i];
END_REPEAT;
IF Mag > 0.0 THEN
    Mag := SQRT(Mag);
    REPEAT i := 1 TO Ndim;
        V.DirectionRatios[i] := V.DirectionRatios[i]/Mag;
    END_REPEAT;
    IF 'IFC4.IFCVECTOR' IN TYPEOF(arg) THEN
        Vec.Orientation := V;
        Result := Vec;
    ELSE
        Result := V;
    END_IF;
ELSE
    RETURN(?);
END_IF;
END_IF;
RETURN (Result);
END_FUNCTION;

FUNCTION IfcOrthogonalComplement
(Vec : IfcDirection)

```

```

        : IfcDirection;
LOCAL
    Result : IfcDirection ;
END_LOCAL;
IF NOT EXISTS (Vec) OR (Vec.Dim <> 2) THEN
    RETURN(?);
ELSE
    Result := IfcRepresentationItem() || IfcGeometricRepresentationItem () || IfcDirection([-
Vec.DirectionRatios[2], Vec.DirectionRatios[1]]);
    RETURN(Result);
END_IF;
END_FUNCTION;

FUNCTION IfcPathHeadToTail
(APath : IfcPath) : LOGICAL;
    LOCAL
        N : INTEGER := 0;
        P : LOGICAL := UNKNOWN;
    END_LOCAL;
    N := SIZEOF (APath.EdgeList);
    REPEAT i := 2 TO N;
        P := P AND (APath.EdgeList[i-1].EdgeEnd :=:
            APath.EdgeList[i].EdgeStart);
    END_REPEAT;
    RETURN (P);
END_FUNCTION;

FUNCTION IfcPointListDim
(PointList : IfcCartesianPointList)
    : IfcDimensionCount;

```

```

IF ('IFC4.IFCCARTESIANPOINTLIST2D' IN TYPEOF(PointList))
    THEN RETURN(2);
END_IF;
IF ('IFC4.IFCCARTESIANPOINTLIST3D' IN TYPEOF(PointList))
    THEN RETURN(3);
END_IF;
RETURN (?);
END_FUNCTION;

FUNCTION IfcSameAxis2Placement
(ap1, ap2 : IfcAxis2Placement; Epsilon : REAL)
    : LOGICAL ;

RETURN (IfcSameDirection(ap1.P[1],ap2.P[1],Epsilon) AND
        IfcSameDirection(ap1.P[2],ap2.P[2],Epsilon) AND
        IfcSameCartesianPoint(ap1.Location,ap2.Location,Epsilon));
END_FUNCTION;

FUNCTION IfcSameCartesianPoint
(cp1, cp2 : IfcCartesianPoint; Epsilon : REAL)
    : LOGICAL;

LOCAL
    cp1x : REAL := cp1.Coordinates[1];
    cp1y : REAL := cp1.Coordinates[2];
    cp1z : REAL := 0;
    cp2x : REAL := cp2.Coordinates[1];
    cp2y : REAL := cp2.Coordinates[2];
    cp2z : REAL := 0;

```

```

END_LOCAL;

IF (SIZEOF(cp1.Coordinates) > 2) THEN
    cp1z := cp1.Coordinates[3];
END_IF;

IF (SIZEOF(cp2.Coordinates) > 2) THEN
    cp2z := cp2.Coordinates[3];
END_IF;

RETURN (IfcSameValue(cp1x, cp2x, Epsilon) AND
        IfcSameValue(cp1y, cp2y, Epsilon) AND
        IfcSameValue(cp1z, cp2z, Epsilon));
END_FUNCTION;

```

```

FUNCTION IfcSameDirection
(dir1, dir2 : IfcDirection; Epsilon : REAL)
    : LOGICAL;
LOCAL
    dir1x : REAL := dir1.DirectionRatios[1];
    dir1y : REAL := dir1.DirectionRatios[2];
    dir1z : REAL := 0;
    dir2x : REAL := dir2.DirectionRatios[1];
    dir2y : REAL := dir2.DirectionRatios[2];
    dir2z : REAL := 0;
END_LOCAL;

IF (SIZEOF(dir1.DirectionRatios) > 2) THEN
    dir1z := dir1.DirectionRatios[3];
END_IF;

```

```

IF (SIZEOF(dir2.DirectionRatios) > 2) THEN
    dir2z := dir2.DirectionRatios[3];
END_IF;

RETURN (IfcSameValue(dir1x,dir2x,Epsilon) AND
        IfcSameValue(dir1y,dir2y,Epsilon) AND
        IfcSameValue(dir1z,dir2z,Epsilon));
END_FUNCTION;

FUNCTION IfcSameValidPrecision
(Epsilon1, Epsilon2 : REAL) : LOGICAL ;
LOCAL
    ValidEps1, ValidEps2 : REAL;
    DefaultEps          : REAL := 0.000001;
    DerivationOfEps     : REAL := 1.001;
    UpperEps            : REAL := 1.0;
END_LOCAL;

    ValidEps1 := NVL(Epsilon1, DefaultEps);
    ValidEps2 := NVL(Epsilon2, DefaultEps);
    RETURN ((0.0 < ValidEps1) AND (ValidEps1 <= (DerivationOfEps * ValidEps2)) AND
            (ValidEps2 <= (DerivationOfEps * ValidEps1)) AND (ValidEps2 < UpperEps));
END_FUNCTION;

FUNCTION IfcSameValue
(Value1, Value2 : REAL; Epsilon : REAL)
: LOGICAL;
LOCAL
    ValidEps : REAL;

```

```

    DefaultEps : REAL := 0.000001;
END_LOCAL;

ValidEps := NVL(Epsilon, DefaultEps);
RETURN ((Value1 + ValidEps > Value2) AND (Value1 < Value2 + ValidEps));
END_FUNCTION;

FUNCTION IfcScalarTimesVector
(Scalar : REAL; Vec : IfcVectorOrDirection)
: IfcVector;
LOCAL
    V : IfcDirection;
    Mag : REAL;
    Result : IfcVector;
END_LOCAL;

IF NOT EXISTS (Scalar) OR NOT EXISTS (Vec) THEN
    RETURN (?);
ELSE
    IF 'IFC4.IFCVECTOR' IN TYPEOF (Vec) THEN
        V := Vec#IfcVector.Orientation;
        Mag := Scalar * Vec#IfcVector.Magnitude;
    ELSE
        V := Vec;
        Mag := Scalar;
    END_IF;
    IF (Mag < 0.0 ) THEN
        REPEAT i := 1 TO SIZEOF(V.DirectionRatios);
            V.DirectionRatios[i] := -V.DirectionRatios[i];
        END_REPEAT;
    END_IF;
END_FUNCTION;

```

```

        Mag := -Mag;
    END_IF;

    Result := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcVector(IfcNormalise(V), Mag);

    END_IF;

    RETURN (Result);

END_FUNCTION;

FUNCTION IfcSecondProjAxis
(ZAxis, XAxis, Arg: IfcDirection)
: IfcDirection;

LOCAL
YAxis : IfcVector;
V : IfcDirection;
Temp : IfcVector;

END_LOCAL;

IF NOT EXISTS(Arg) THEN
    V := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([0.0,1.0,0.0]);
ELSE
    V := Arg;
END_IF;

Temp := IfcScalarTimesVector(IfcDotProduct(V, ZAxis), ZAxis);
YAxis := IfcVectorDifference(V, Temp);

Temp := IfcScalarTimesVector(IfcDotProduct(V, XAxis), XAxis);
YAxis := IfcVectorDifference(YAxis, Temp);

YAxis := IfcNormalise(YAxis);

RETURN(YAxis.Orientation);

END_FUNCTION;

```

```
FUNCTION IfcShapeRepresentationTypes
```

```
(RepType : IfcLabel; Items : SET OF IfcRepresentationItem) : LOGICAL;
```

```
LOCAL
```

```
    Count : INTEGER := 0;
```

```
END_LOCAL;
```

```
CASE RepType OF
```

```
'Point' :
```

```
    BEGIN
```

```
        Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCPOINT' IN TYPEOF(temp))));
```

```
    END;
```

```
'PointCloud' :
```

```
    BEGIN
```

```
        Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCCARTESIANPOINTLIST3D' IN TYPEOF(temp))));
```

```
    END;
```

```
'Curve' :
```

```
    BEGIN
```

```
        Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCCURVE' IN TYPEOF(temp))));
```

```
    END;
```

```
'Curve2D' :
```

```
    BEGIN
```

```
        Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCCURVE' IN TYPEOF(temp))  
            AND (tempWIfcCurve.Dim = 2)));
```

```
    END;
```



```

'Curve3D' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCCURVE' IN TYPEOF(temp))
        AND (tempWIfcCurve.Dim = 3)));
END;

'Surface' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCSURFACE' IN TYPEOF(temp))));
END;

'Surface2D' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCSURFACE' IN TYPEOF(temp))
        AND (tempWIfcSurface.Dim = 2)));
END;

'Surface3D' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCSURFACE' IN TYPEOF(temp))
        AND (tempWIfcSurface.Dim = 3)));
END;

'FillArea' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCANNOTATIONFILLAREA' IN TYPEOF(temp))));
END;

'Text' :
BEGIN

```

```
Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCTEXTLITERAL' IN TYPEOF(temp))));  
END;
```

```
'AdvancedSurface' :
```

```
BEGIN  
Count := SIZEOF(QUERY(temp <* Items | 'IFC4.IFCBSPLINESURFACE' IN TYPEOF(temp))));  
END;
```

```
'Annotation2D' :
```

```
BEGIN  
Count := SIZEOF(QUERY(temp <* Items | (  
SIZEOF(TYPEOF(temp) * [  
'IFC4.IFCPOINT',  
'IFC4.IFCCURVE',  
'IFC4.IFCGEOMETRICCURVESET',  
'IFC4.IFCANNOTATIONFILLAREA',  
'IFC4.IFCTEXTLITERAL']) = 1)  
));  
END;
```

```
'GeometricSet' :
```

```
BEGIN  
Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCGEOMETRICSET' IN TYPEOF(temp))  
OR ('IFC4.IFCPOINT' IN TYPEOF(temp))  
OR ('IFC4.IFCCURVE' IN TYPEOF(temp))  
OR ('IFC4.IFCSURFACE' IN TYPEOF(temp))));  
END;
```

```
'GeometricCurveSet' :
```

```
BEGIN
```

```

Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCGEOMETRICCURVESET' IN TYPEOF(temp))
    OR ('IFC4.IFCGEOMETRICSET' IN TYPEOF(temp))
    OR ('IFC4.IFCPOINT' IN TYPEOF(temp))
    OR ('IFC4.IFCCURVE' IN TYPEOF(temp))));
REPEAT i:=1 TO HIINDEX(Items);
    IF ('IFC4.IFCGEOMETRICSET' IN TYPEOF(Items[i]))
    THEN
        IF (SIZEOF(QUERY(temp <* Items[i]WIfcGeometricSet.Elements | 'IFC4.IFCSURFACE' IN
TYPEOF(temp))) > 0)
        THEN
            Count := Count - 1;
        END_IF;
    END_IF;
END_REPEAT;
END;

```

'Tessellation' :

```

BEGIN
    Count := SIZEOF(QUERY(temp <* Items | 'IFC4.IFCTESSELLATEDITEM' IN TYPEOF(temp)));
END;

```

'SurfaceOrSolidModel' :

```

BEGIN
    Count := SIZEOF(QUERY(temp <* Items | SIZEOF([
        'IFC4.IFCTESSELLATEDITEM',
        'IFC4.IFCSHELLBASEDSURFACEMODEL',
        'IFC4.IFCFACEBASEDSURFACEMODEL',
        'IFC4.IFCSOLIDMODEL'] * TYPEOF(temp)) >= 1
    ));
END;

```

'SurfaceModel' :

BEGIN

```
Count := SIZEOF(QUERY(temp <* Items | SIZEOF([
    'IFC4.IFCTESSELLATEDITEM',
    'IFC4.IFCSHELLBASEDSURFACEMODEL',
    'IFC4.IFCFACEBASEDSURFACEMODEL'] * TYPEOF(temp)) >= 1
));
```

END;

'SolidModel' :

BEGIN

```
Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCSOLIDMODEL' IN TYPEOF(temp))));
```

END;

'SweptSolid' :

BEGIN

```
Count := SIZEOF(QUERY(temp <* Items | (SIZEOF([
    'IFC4.IFCEXTRUDEDAREASOLID',
    'IFC4.IFCREVOLVEDAREASOLID'] * TYPEOF(temp)) >= 1
) AND (SIZEOF([
    'IFC4.IFCEXTRUDEDAREASOLIDTAPERED',
    'IFC4.IFCREVOLVEDAREASOLIDTAPERED'] * TYPEOF(temp)) = 0
)
));
```

END;

'AdvancedSweptSolid' :

BEGIN

```
Count := SIZEOF(QUERY(temp <* Items | SIZEOF([
```

```

        'IFC4.IFCSWEPTAREASOLID',
        'IFC4.IFCSWEPTDISKSOLID'] * TYPEOF(temp)) >= 1
    ));

END;

'CSG' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | SIZEOF([
        'IFC4.IFCBOOLEANRESULT',
        'IFC4.IFCCSGPRIMITIVE3D',
        'IFC4.IFCCSGSOLID'] * TYPEOF(temp)) >= 1
    ));

END;

'Clipping' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | SIZEOF([
        'IFC4.IFCCSGSOLID',
        'IFC4.IFCBOOLEANCLIPPINGRESULT'] * TYPEOF(temp)) >= 1
    ));

END;

'Brep' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCFACETEDBREP' IN TYPEOF(temp))));

END;

'AdvancedBrep' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCMANIFOLDSOLIDBREP' IN TYPEOF(temp))));

```

```

END;

'BoundingBox' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCBOUNDINGBOX' IN TYPEOF(temp))));
    IF (SIZEOF(Items) > 1)
    THEN
        Count := 0;
    END_IF;
END;

'SectionedSpine' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCSECTIONEDSPINE' IN TYPEOF(temp))));
END;

'LightSource' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCLIGHTSOURCE' IN TYPEOF(temp))));
END;

'MappedRepresentation' :
BEGIN
    Count := SIZEOF(QUERY(temp <* Items | ('IFC4.IFCMAPPEDITEM' IN TYPEOF(temp))));
END;

OTHERWISE : RETURN(?);
END_CASE;
RETURN (Count = SIZEOF(Items));
END_FUNCTION;

```

```

FUNCTION IfcSurfaceWeightsPositive
( B: IfcRationalBSplineSurfaceWithKnots)
: BOOLEAN;

LOCAL
    Result : BOOLEAN := TRUE;
END_LOCAL;

REPEAT i := 0 TO B.IfcbSplineSurface.UUpper;
    REPEAT j := 0 TO B.IfcbSplineSurface.VUpper;
        IF (B.Weights[i][j] <= 0.0) THEN
            Result := FALSE;
            RETURN(Result);
        END_IF;
    END_REPEAT;
END_REPEAT;
RETURN(Result);
END_FUNCTION;

```

```

FUNCTION IfcTaperedSweptAreaProfiles
(StartArea, EndArea : IfcProfileDef)
: LOGICAL;

```

```

LOCAL
    Result : LOGICAL := FALSE;
END_LOCAL;

```

```

IF ('IFC4.IFCPARAMETERIZEDPROFILEDEF' IN TYPEOF(StartArea)) THEN
    IF ('IFC4.IFCDERIVEDPROFILEDEF' IN TYPEOF(EndArea)) THEN

```

```

        Result := (StartArea :=: EndAreaWlfcDerivedProfileDef.ParentProfile);
ELSE
    Result := (TYPEOF(StartArea) = TYPEOF(EndArea));
END_IF;
ELSE
    IF ('IFC4.IFCDERIVEDPROFILEDEF' IN TYPEOF(EndArea)) THEN
        Result := (StartArea :=: EndAreaWlfcDerivedProfileDef.ParentProfile);
    ELSE
        Result := FALSE;
    END_IF;
END_IF;

RETURN(Result);
END_FUNCTION;

```

```

FUNCTION IfcTopologyRepresentationTypes
(RepType : IfcLabel; Items : SET OF IfcRepresentationItem) : LOGICAL;

```

```

    LOCAL
        Count : INTEGER := 0;
    END_LOCAL;

    CASE RepType OF
        'Vertex' :
            BEGIN
                Count := SIZEOF(QUERY(temp <* Items |
                    ('IFC4.IFCVERTEX' IN TYPEOF(temp))));
            END;
        'Edge' :
            BEGIN

```



```

        Count := SIZEOF(QUERY(temp <* Items |
                               ('IFC4.IFCEDGE' IN TYPEOF(temp))));
    END;
'Path' :
    BEGIN
        Count := SIZEOF(QUERY(temp <* Items |
                               ('IFC4.IFCPATH' IN TYPEOF(temp))));
    END;
'Face' :
    BEGIN
        Count := SIZEOF(QUERY(temp <* Items |
                               ('IFC4.IFCFACE' IN TYPEOF(temp))));
    END;
'Shell' :
    BEGIN
        Count := SIZEOF(QUERY(temp <* Items |
                               ('IFC4.IFCOPENSHELL' IN TYPEOF(temp))
                               OR ('IFC4.IFCCLOSEDSHELL' IN TYPEOF(temp))));
    END;
'Undefined' : RETURN(TRUE);
    OTHERWISE : RETURN(?);
END_CASE;
RETURN (Count = SIZEOF(Items));
END_FUNCTION;

FUNCTION IfcUniqueDefinitionNames
(Relations : SET [1:?] OF IfcReIDefinesByProperties)
:LOGICAL;

LOCAL

```

```

Definition : IfcPropertySetDefinitionSelect;
DefinitionSet : IfcPropertySetDefinitionSet;
Properties : SET OF IfcPropertySetDefinition := [];
Result : LOGICAL;
END_LOCAL;

IF SIZEOF(Relations) = 0 THEN
    RETURN(TRUE);
END_IF;

REPEAT i:=1 TO HIINDEX(Relations);
    Definition := Relations[i].RelatingPropertyDefinition;
    IF 'IFC4.IFCPROPERTYSETDEFINITION' IN TYPEOF(Definition) THEN
        Properties := Properties + Definition;
    ELSE
        IF 'IFC4.IFCPROPERTYSETDEFINITIONSET' IN TYPEOF(Definition) THEN
            BEGIN
                DefinitionSet := Definition;
                REPEAT j:= 1 TO HIINDEX(DefinitionSet);
                    Properties := Properties + DefinitionSet[j];
                END_REPEAT;
            END;
        END_IF;
    END_IF;
END_REPEAT;

Result := IfcUniquePropertySetNames(Properties);
RETURN (Result);
END_FUNCTION;

```

```

FUNCTION IfcUniquePropertyName
(Properties : SET [1:?] OF IfcProperty)
:LOGICAL;

LOCAL
  Names : SET OF IfcIdentifier := [];
END_LOCAL;

REPEAT i:=1 TO HIINDEX(Properties);
  Names := Names + Properties[i].Name;
END_REPEAT;

RETURN (SIZEOF(Names) = SIZEOF(Properties));
END_FUNCTION;

```

```

FUNCTION IfcUniquePropertySetNames
(Properties : SET [1:?] OF IfcPropertySetDefinition)
:LOGICAL;

LOCAL
  Names : SET OF IfcLabel := [];
  Unnamed : INTEGER := 0;
END_LOCAL;

REPEAT i:=1 TO HIINDEX(Properties);
  IF 'IFC4.IFCPROPERTYSET' IN TYPEOF(Properties[i]) THEN
    Names := Names + Properties[i].IfcRoot.Name;
  ELSE
    Unnamed := Unnamed + 1;
  END_IF;
END_REPEAT;

```

END_REPEAT;

RETURN (SIZEOF(Names) + Unnamed = SIZEOF(Properties));

END_FUNCTION;

FUNCTION IfcUniquePropertyTemplateNameNames

(Properties : SET [1:?] OF IfcPropertyTemplate)

:LOGICAL;

LOCAL

Names : SET OF IfcLabel := [];

END_LOCAL;

REPEAT i:=1 TO HIINDEX(Properties);

Names := Names + Properties[i].Name;

END_REPEAT;

RETURN (SIZEOF(Names) = SIZEOF(Properties));

END_FUNCTION;

FUNCTION IfcUniqueQuantityNames

(Properties : SET [1:?] OF IfcPhysicalQuantity)

:LOGICAL;

LOCAL

Names : SET OF IfcLabel := [];

END_LOCAL;

REPEAT i:=1 TO HIINDEX(Properties);

Names := Names + Properties[i].Name;

END_REPEAT;

```
RETURN (SIZEOF(NAMES) = SIZEOF(Properties));  
END_FUNCTION;
```

```
FUNCTION IfcVectorDifference  
(Arg1, Arg2 : IfcVectorOrDirection)  
  : IfcVector;
```

```
LOCAL
```

```
  Result : IfcVector;  
  Res, Vec1, Vec2 : IfcDirection;  
  Mag, Mag1, Mag2 : REAL;  
  Ndim : INTEGER;
```

```
END_LOCAL;
```

```
IF ((NOT EXISTS (Arg1)) OR (NOT EXISTS (Arg2))) OR (Arg1.Dim <> Arg2.Dim) THEN
```

```
  RETURN (?);
```

```
ELSE
```

```
  BEGIN
```

```
    IF 'IFC4.IFCVECTOR' IN TYPEOF(Arg1) THEN
```

```
      Mag1 := Arg1.IfVector.Magnitude;
```

```
      Vec1 := Arg1.IfVector.Orientation;
```

```
    ELSE
```

```
      Mag1 := 1.0;
```

```
      Vec1 := Arg1;
```

```
    END_IF;
```

```
    IF 'IFC4.IFCVECTOR' IN TYPEOF(Arg2) THEN
```

```
      Mag2 := Arg2.IfVector.Magnitude;
```

```
      Vec2 := Arg2.IfVector.Orientation;
```

```
    ELSE
```

```
      Mag2 := 1.0;
```

```
      Vec2 := Arg2;
```

```

    END_IF;

    Vec1 := IfcNormalise (Vec1);
    Vec2 := IfcNormalise (Vec2);
    Ndim := SIZEOF(Vec1.DirectionRatios);
    Mag := 0.0;
    Res := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([0.0:Ndim]);

    REPEAT i := 1 TO Ndim;
        Res.DirectionRatios[i] := Mag1*Vec1.DirectionRatios[i] - Mag2*Vec2.DirectionRatios[i];
        Mag := Mag + (Res.DirectionRatios[i]*Res.DirectionRatios[i]);
    END_REPEAT;

    IF (Mag > 0.0 ) THEN
        Result := IfcRepresentationItem() || IfcGeometricRepresentationItem () || IfcVector( Res,
SQRT(Mag));
    ELSE
        Result := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcVector( Vec1, 0.0);
    END_IF;
END;
END_IF;
RETURN (Result);
END_FUNCTION;

FUNCTION IfcVectorSum
(Arg1, Arg2 : IfcVectorOrDirection)
: IfcVector;
LOCAL
Result : IfcVector;
Res, Vec1, Vec2 : IfcDirection;

```

```

Mag, Mag1, Mag2 : REAL;
Ndim : INTEGER;
END_LOCAL;

IF ((NOT EXISTS (Arg1)) OR (NOT EXISTS (Arg2))) OR (Arg1.Dim <> Arg2.Dim) THEN
    RETURN (?);
ELSE
    BEGIN
        IF 'IFC4.IFCVECTOR' IN TYPEOF(Arg1) THEN
            Mag1 := Arg1WIfcVector.Magnitude;
            Vec1 := Arg1WIfcVector.Orientation;
        ELSE
            Mag1 := 1.0;
            Vec1 := Arg1;
        END_IF;
        IF 'IFC4.IFCVECTOR' IN TYPEOF(Arg2) THEN
            Mag2 := Arg2WIfcVector.Magnitude;
            Vec2 := Arg2WIfcVector.Orientation;
        ELSE
            Mag2 := 1.0;
            Vec2 := Arg2;
        END_IF;
        Vec1 := IfcNormalise (Vec1);
        Vec2 := IfcNormalise (Vec2);
        Ndim := SIZEOF(Vec1.DirectionRatios);
        Mag := 0.0;
        Res := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcDirection([0.0:Ndim]);

        REPEAT i := 1 TO Ndim;

```

```

    Res.DirectionRatios[i] := Mag1*Vec1.DirectionRatios[i] + Mag2*Vec2.DirectionRatios[i];
    Mag := Mag + (Res.DirectionRatios[i]*Res.DirectionRatios[i]);
END_REPEAT;

IF (Mag > 0.0 ) THEN
    Result := IfcRepresentationItem() || IfcGeometricRepresentationItem () || IfcVector( Res,
SQRT(Mag));
ELSE
    Result := IfcRepresentationItem() || IfcGeometricRepresentationItem () ||
IfcVector( Vec1, 0.0);
END_IF;
END;
END_IF;
RETURN (Result);
END_FUNCTION;

END_SCHEMA;

```