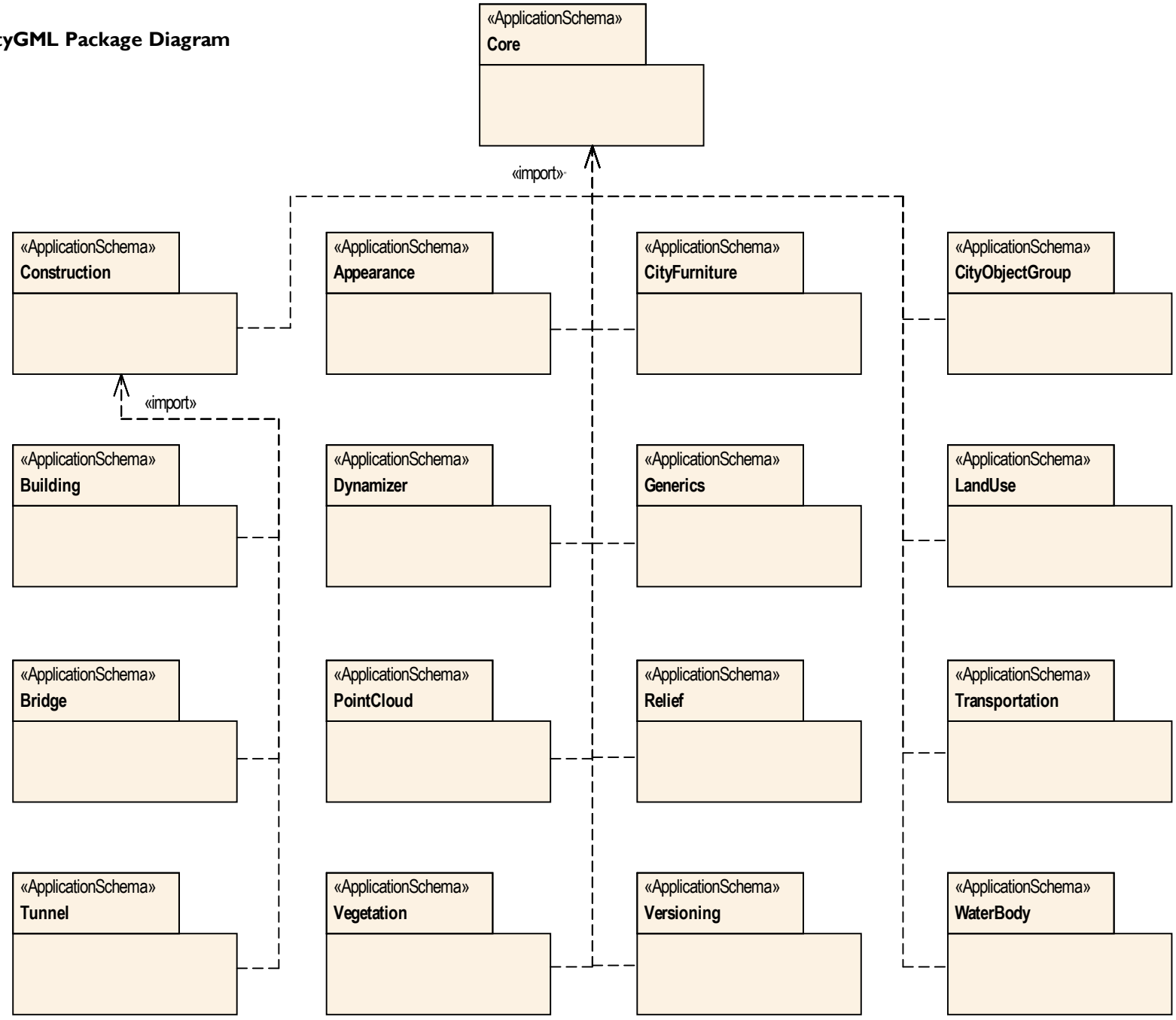
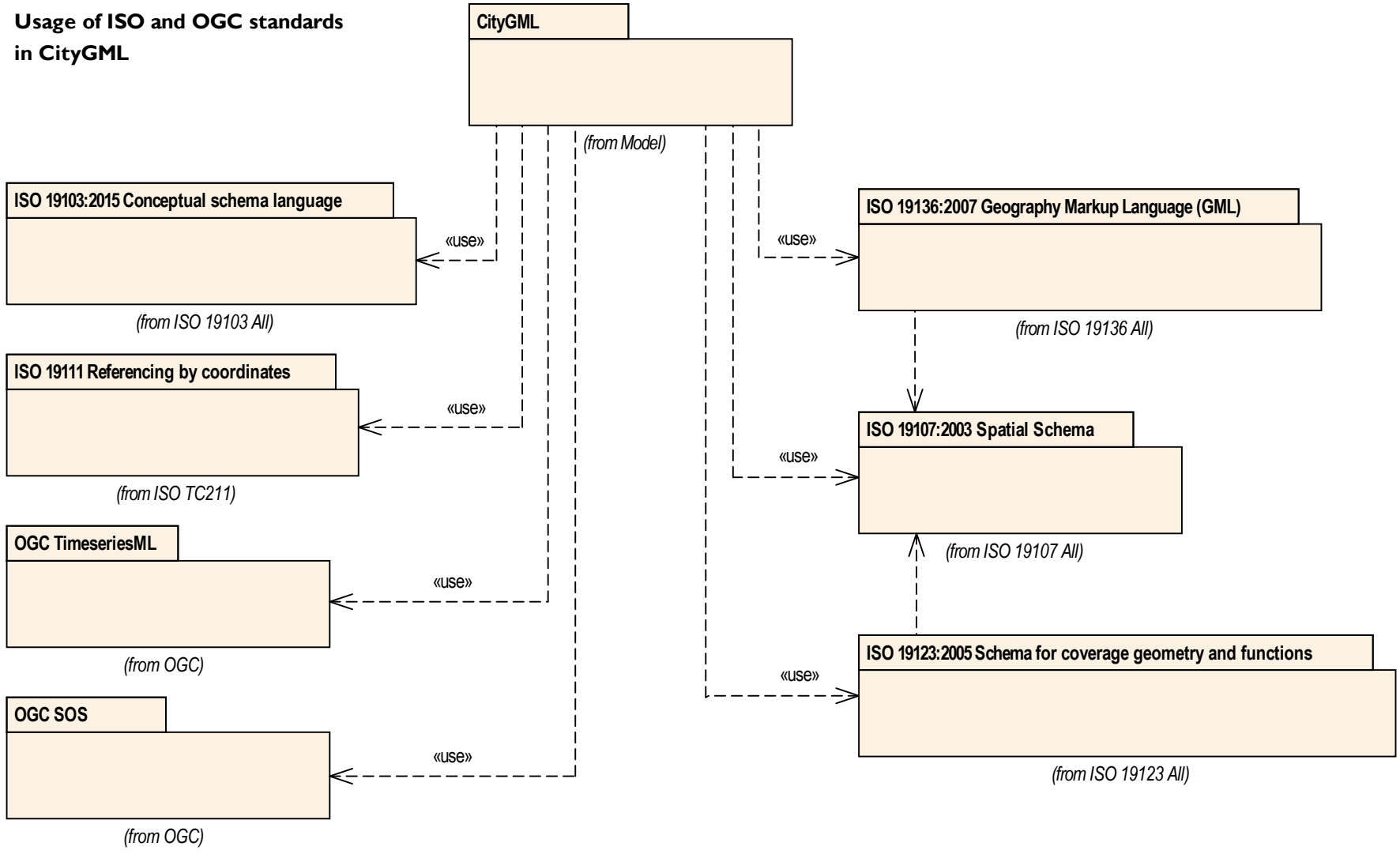
CityGML Package Diagram



Usage of ISO and OGC standards in CityGML



Core module - Overview

«FeatureType»
AbstractFeatureWithLifespan
«Property»
+ creationDate: DateTime [0..1]
+ terminationDate: DateTime [0..1]
+ validFrom: DateTime [0..1]
+ validTo: DateTime [0..1]

The class *AbstractFeatureWithLifespan* has the stereotype <<*FeatureType*>>. When encoding the class in a GML application schema, it inherits the attributes description, descriptionReference, name, identifier, and boundedBy from the classes *AbstractGML* and *AbstractFeature*.

+featureMember «Property» * — «FeatureType» ***General Feature Model:: AnyFeature***

«FeatureType» ***AbstractVersion***
«FeatureType» ***AbstractVersionTransition***
«FeatureType» ***AbstractAppearance***
«FeatureType» ***AbstractDynamizer***
+appearance «Property» *
+dynamizer «Property» *
0..* +versionMember «Property»
0..* +versionTransitionMember «Property»
+appearance «Property» * *
+appearanceMember «Property» *

«DataType» **ExternalReference**
«Property»
+ targetResource: URI
+ informationSystem: URI [0..1]
+ relationType: URI [0..1]

+externalReference «Property» 1 — *

«FeatureType» **CityModel**
«Property»
+ engineeringCRS: EngineeringCRS [0..1]

«FeatureType» ***AbstractCityObject***
«Property»
+ relativeToTerrain: RelativeToTerrain [0..1]
+ relativeToWater: RelativeToWater [0..1]

+cityObjectMember «Property» * *
+generalizesTo «Property» *
+relatedTo «Property» *
0..1 +genericAttribute «Property» *

«DataType» **AbstractGenericAttribute**
«Property»
+ name: CharacterString

CityObjectRelation
«Property»
+ relationType: RelationTypeValue

+genericAttribute «Property» * — 0..1

GM_Primitive «type» **Geometric primitive:: GM_Solid**
0..1 +lod1Solid «Property» *
+lod2Solid «Property» *
+lod3Solid «Property» *

«FeatureType» ***AbstractSpace***
«Property»
+ spaceType: SpaceType [0..1]
+ volume: QualifiedVolume [0..*]
+ area: QualifiedArea [0..*]

+bounds «Property» * — +lod0MultiSurface «Property» *
+lod1MultiSurface «Property» *
+lod2MultiSurface «Property» *
+lod3MultiSurface «Property» *

GM_MultiPrimitive «type» **Geometric aggregates:: GM_MultiSurface**
0..1

+boundary «Property» * — «FeatureType» ***AbstractThematicSurface***
«Property»
+ area: QualifiedArea [0..*]

«DataType» **QualifiedVolume**
«Property»
+ volume: Volume
+ typeOfVolume: QualifiedVolumeValue

«DataType» **QualifiedArea**
«Property»
+ area: Area
+ typeOfArea: QualifiedAreaValue

GM_Primitive «type» **Geometric primitive:: GM_Point**
0..1 +lod0Point «Property» *
1 +referencePoint «Property»

+lod2MultiCurve «Property» *
+lod3MultiCurve «Property» *

GM_MultiPrimitive «type» **Geometric aggregates:: GM_MultiCurve**
0..1
+lod0MultiCurve «Property» *

«type» ***Geometry root:: GM_Object***
+relativeGMLGeometry «Property» 0..1 *

+lod1TerrainIntersectionCurve «Property» *
+lod2TerrainIntersectionCurve «Property» *
+lod3TerrainIntersectionCurve «Property» *

«ObjectType» **ImplicitGeometry**
«Property»
+ transformationMatrix: TransformationMatrix4x4
+ mimeType: MimeValue [0..1]
+ libraryObject: URI [0..1]

«FeatureType» ***AbstractLogicalSpace***
«FeatureType» ***AbstractPhysicalSpace***
+pointCloud «Property» 0..1 — 0..1 +pointCloud «Property»
«FeatureType» ***AbstractPointCloud***

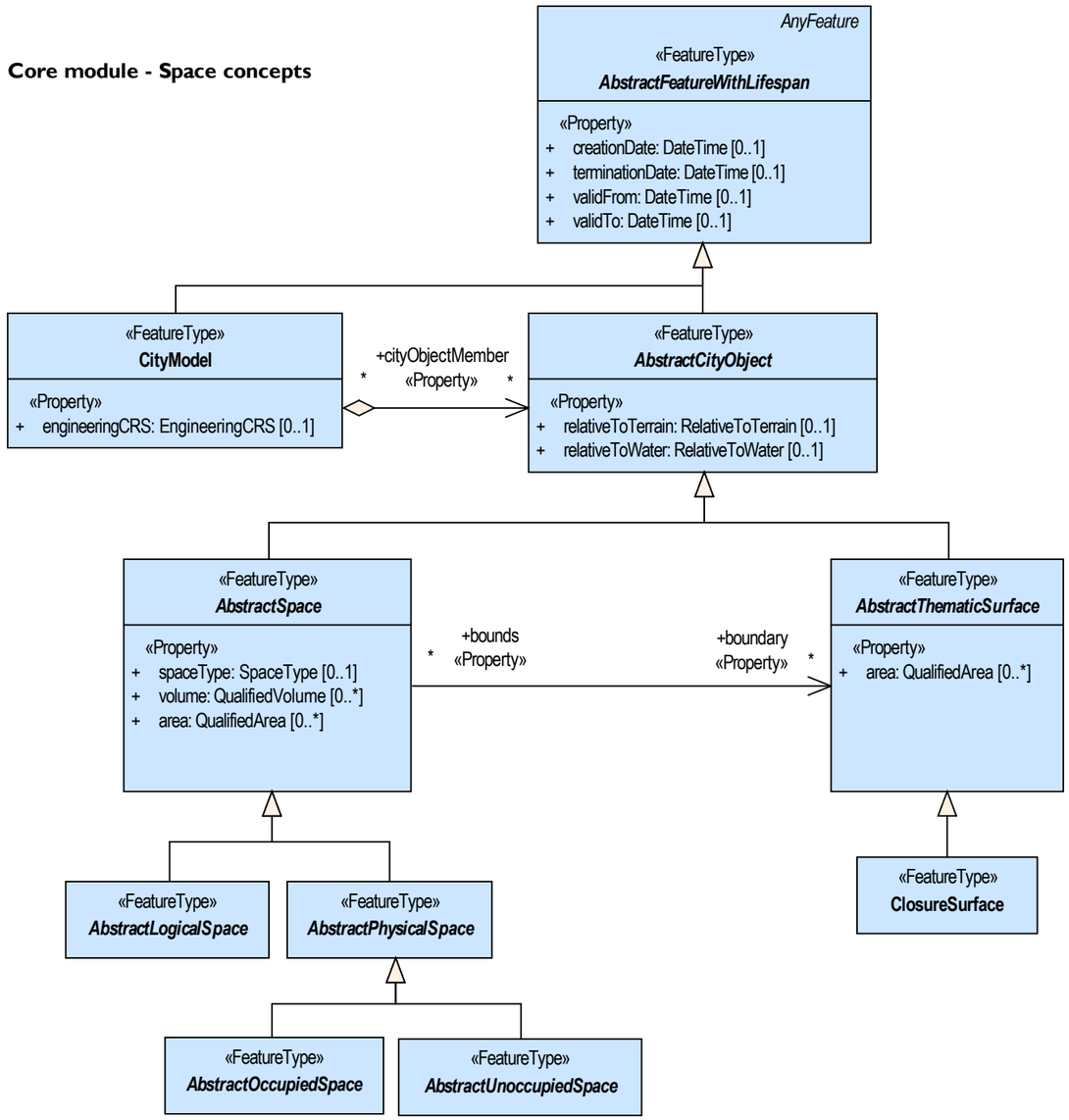
«FeatureType» **ClosureSurface**

«FeatureType» ***AbstractOccupiedSpace***
«FeatureType» ***AbstractUnoccupiedSpace***

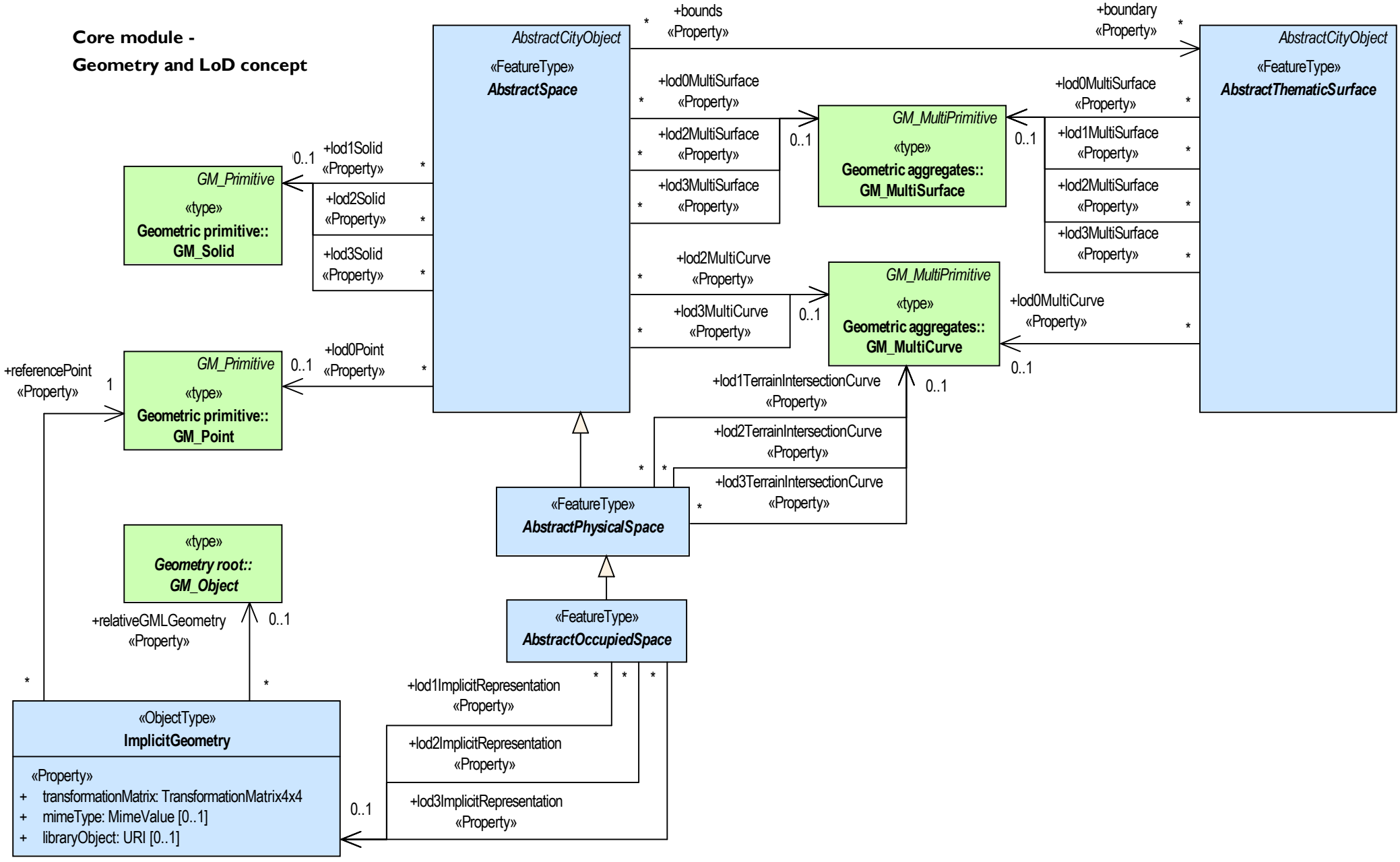
0..1 +lod1ImplicitRepresentation «Property» * *
+lod2ImplicitRepresentation «Property» *
+lod3ImplicitRepresentation «Property» *

GM_MultiPrimitive «type» **Geometric aggregates:: GM_MultiPoint**
+multiPoint «Property» 0..1 — * «FeatureType» **Address** +xalAddress «Property» — 1 «DataType» **XALAddressDetails**

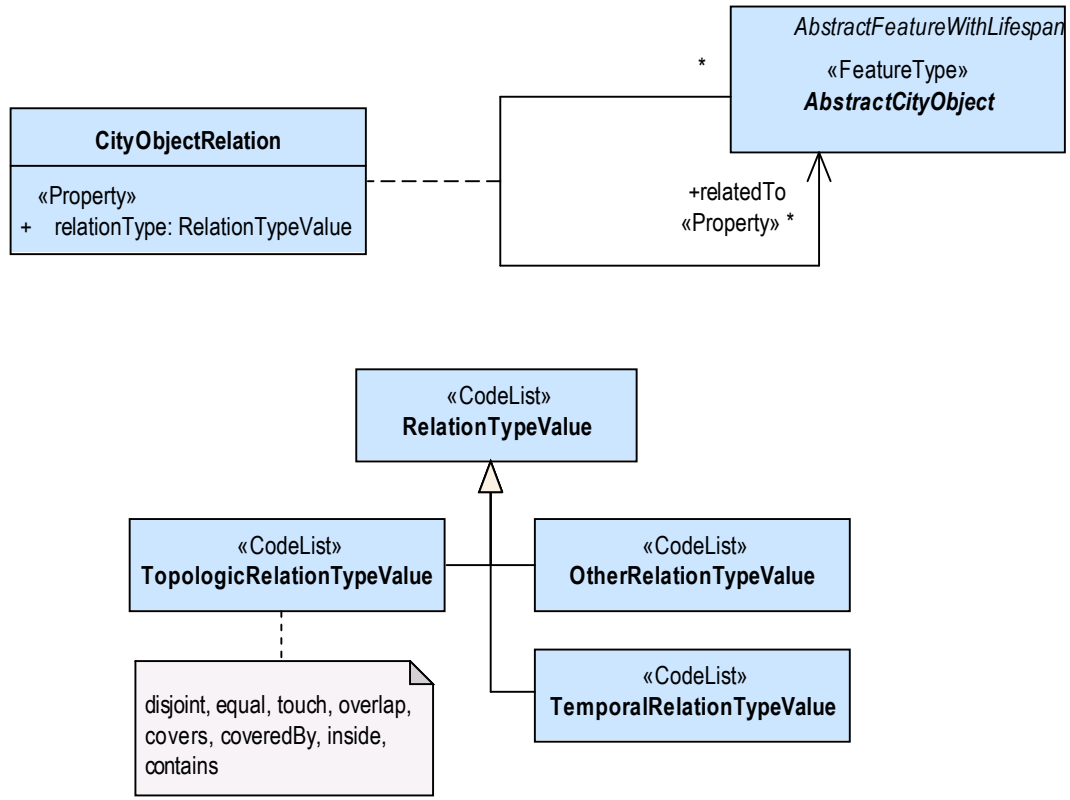
Core module - Space concepts



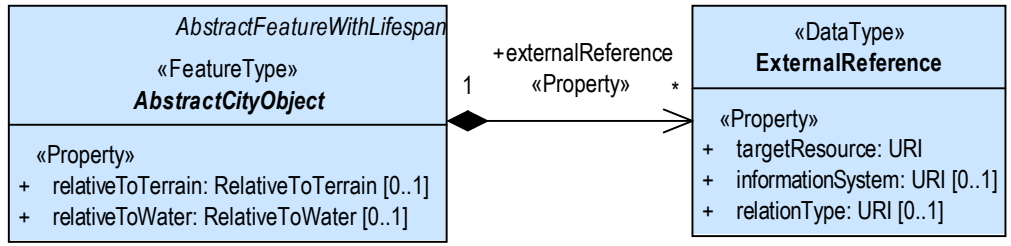
Core module -
Geometry and LoD concept



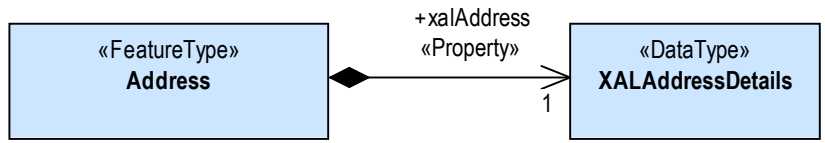
Core module - City object relations



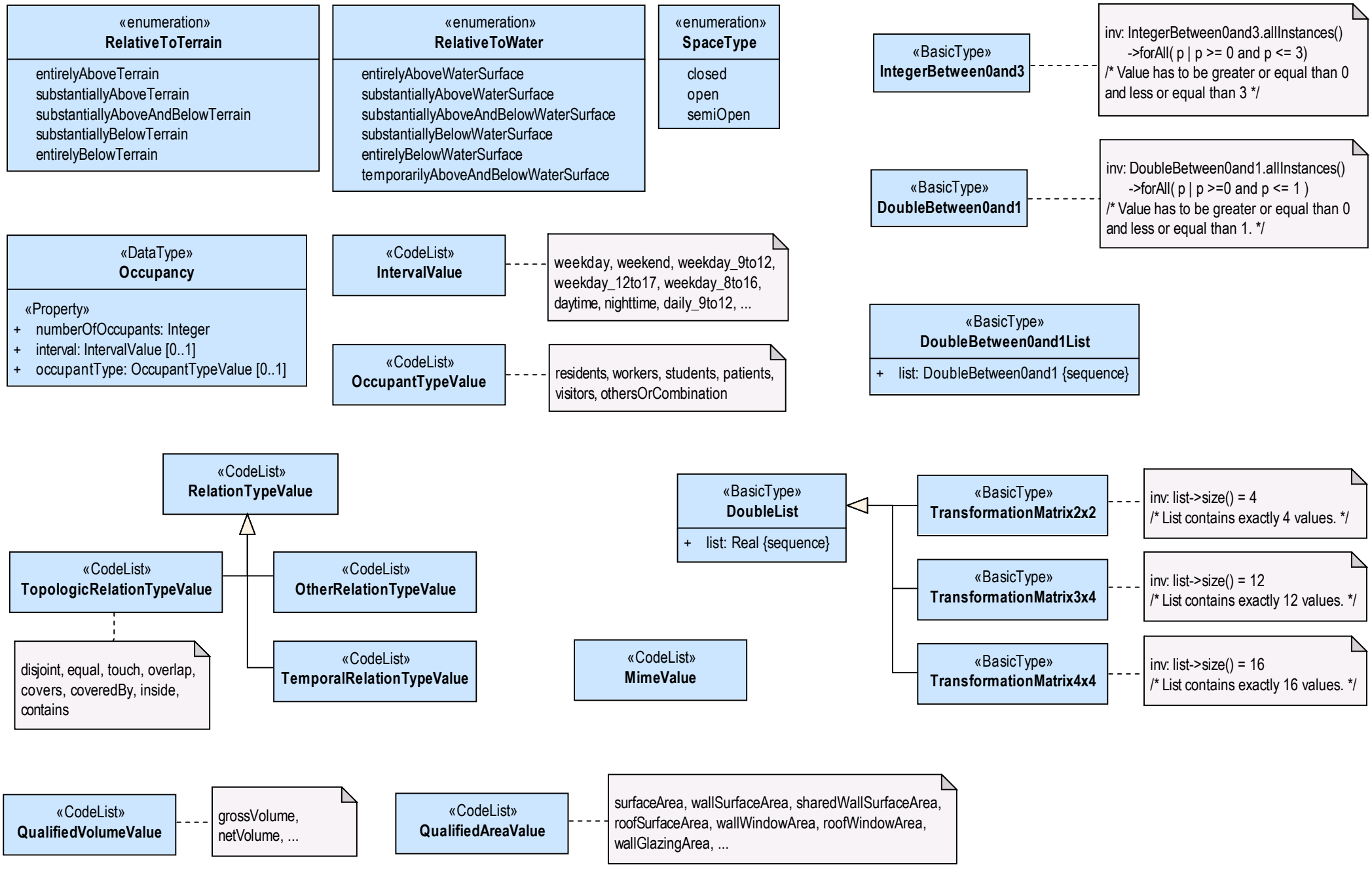
Core module - Miscellaneous



ExternalReference is now extended by an optional ***relationType*** which can link to some external definition of the type of relation (e.g. the *sameAs* relation from OWL). Hence, ***ExternalReferences*** can now be used to express relations similar to RDF.



Core module - Basic Types, Enumerations, and Code lists



Appearance module

AnyFeature
«FeatureType»
Core::
AbstractFeatureWithLifespan

«FeatureType»
Core::CityModel

+cityObjectMember
«Property»

«FeatureType»
Core::AbstractCityObject

+appearance
«Property»

«FeatureType»
Core::AbstractAppearance

+appearanceMember
«Property»

«ObjectType»
Core::ImplicitGeometry

+appearance
«Property»
0..1

«FeatureType»
AbstractSurfaceData
«Property»
+ isFront: Boolean [0..1] = true

+surfaceData
«Property»

«FeatureType»
Appearance
«Property»
+ theme: CharacterString [0..1]

«FeatureType»
X3DMaterial
«Property»
+ ambientIntensity: DoubleBetween0and1 [0..1] = 0.2
+ diffuseColor: Color [0..1] = 0.8 0.8 0.8
+ emissiveColor: Color [0..1] = 0.0 0.0 0.0
+ specularColor: Color [0..1] = 1.0 1.0 1.0
+ shininess: DoubleBetween0and1 [0..1] = 0.2
+ transparency: DoubleBetween0and1 [0..1] = 0.0
+ isSmooth: Boolean [0..1] = false
+ target: URI [0..*]

«FeatureType»
AbstractTexture
«Property»
+ imageURI: URI
+ mimeType: Code [0..1]
+ textureType: TextureType [0..1]
+ wrapMode: WrapMode [0..1]
+ borderColor: ColorPlusOpacity [0..1]

+referencePoint
«Property»
1

GM_Primitive
«type»
Geometric primitive::
GM_Point

+referencePoint
«Property»
0..1

«FeatureType»
ParameterizedTexture

«FeatureType»
GeoreferencedTexture
«Property»
+ preferWorldFile: Boolean [0..1] = true
+ orientation: TransformationMatrix2x2 [0..1]
+ target: URI [0..*]

«ObjectType»
TextureAssociation
«Property»
+ uri: URI

+textureParameterization
«Property»

«DataType»
AbstractTextureParameterization

IO_IdentifiedObjectBase
RS_ReferenceSystem
«type»
Coordinate Reference
Systems::SC_CRS

+crs
«Property»
0..1

«DataType»
TexCoordGen
«Property»
+ worldToTexture: TransformationMatrix3x4

«DataType»
TexCoordList
«Property»
+ textureCoordinates: DoubleList [1..*]
+ ring: URI [1..*]

«BasicType»
Core::DoubleBetween0and1List
+ list: DoubleBetween0and1 {sequence}

«BasicType»
Color

inv: list->size() = 3
/* List contains exactly 3 values. */

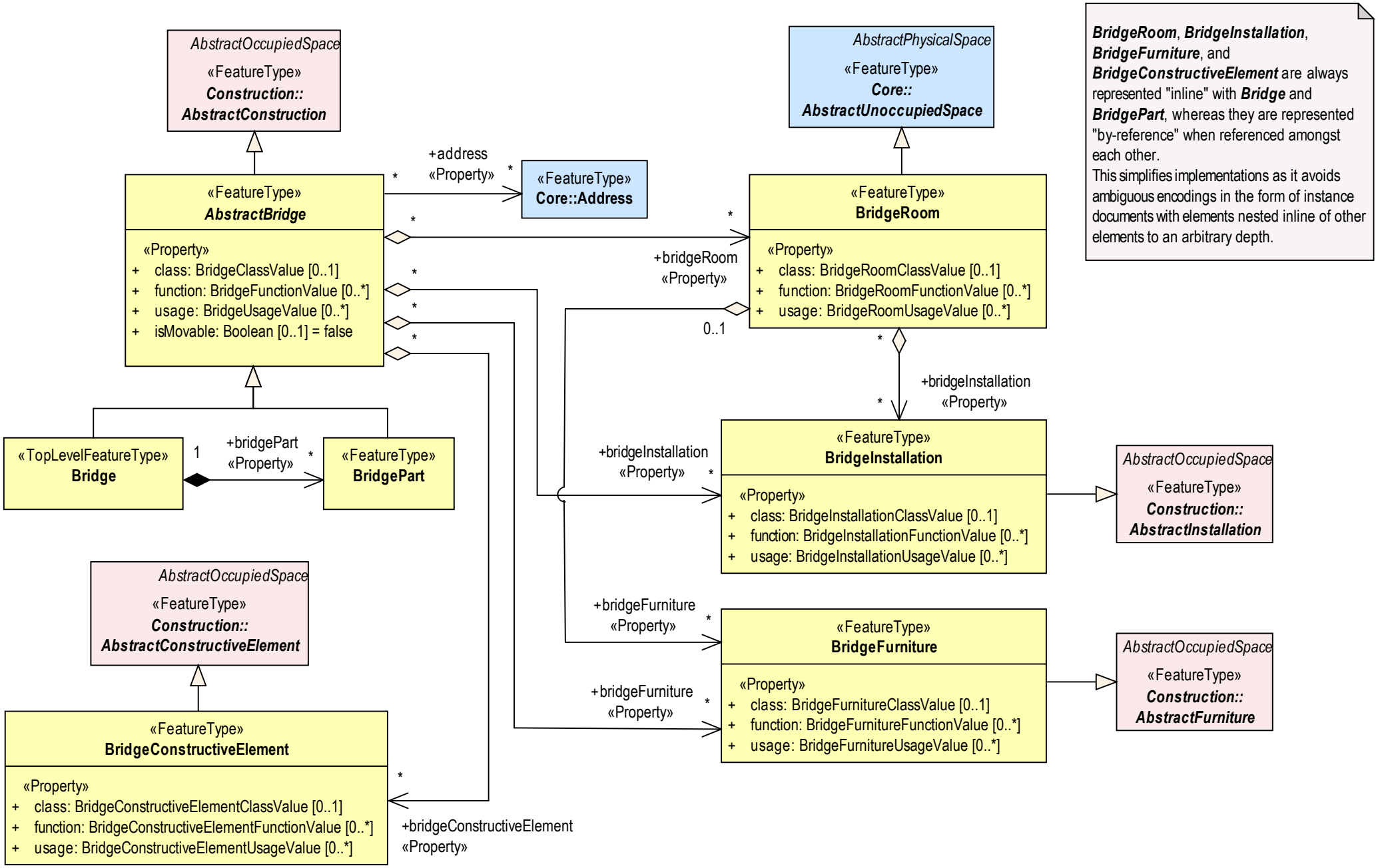
«BasicType»
ColorPlusOpacity

inv: list->size() = 3 or list->size() = 4
/* List contains exactly 3 to 4 values. */

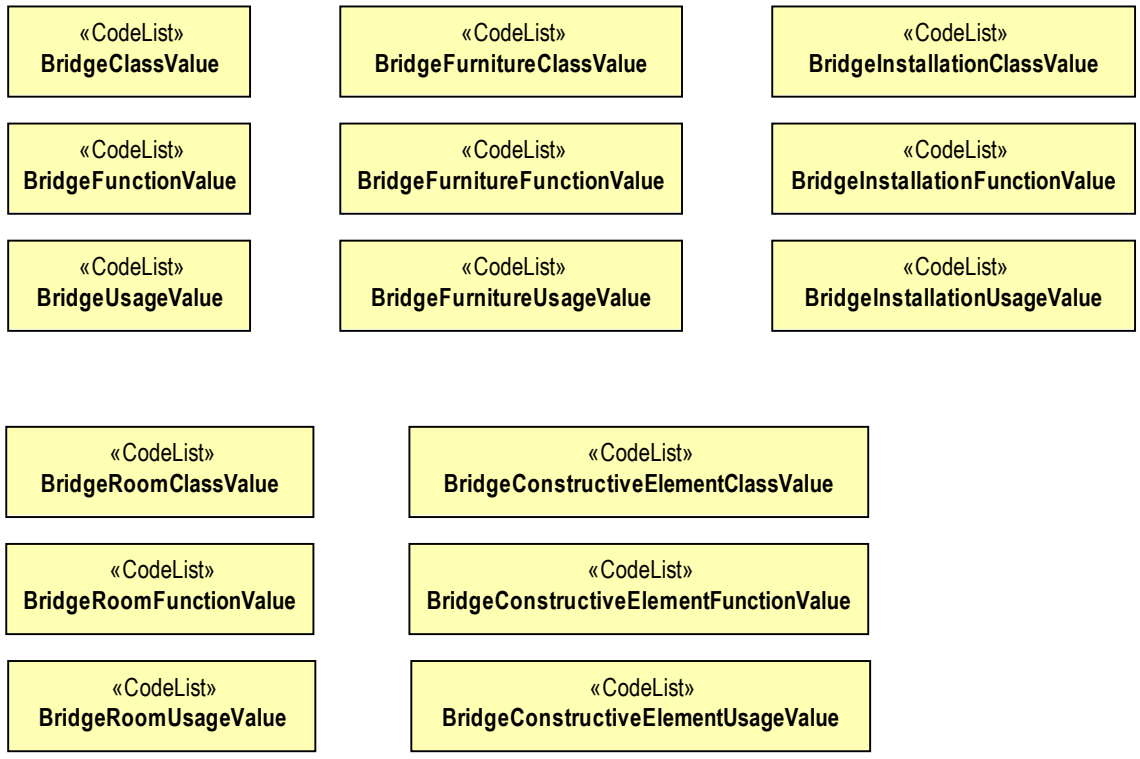
«enumeration»
TextureType
specific
typical
unknown

«enumeration»
WrapMode
none
wrap
mirror
clamp
border

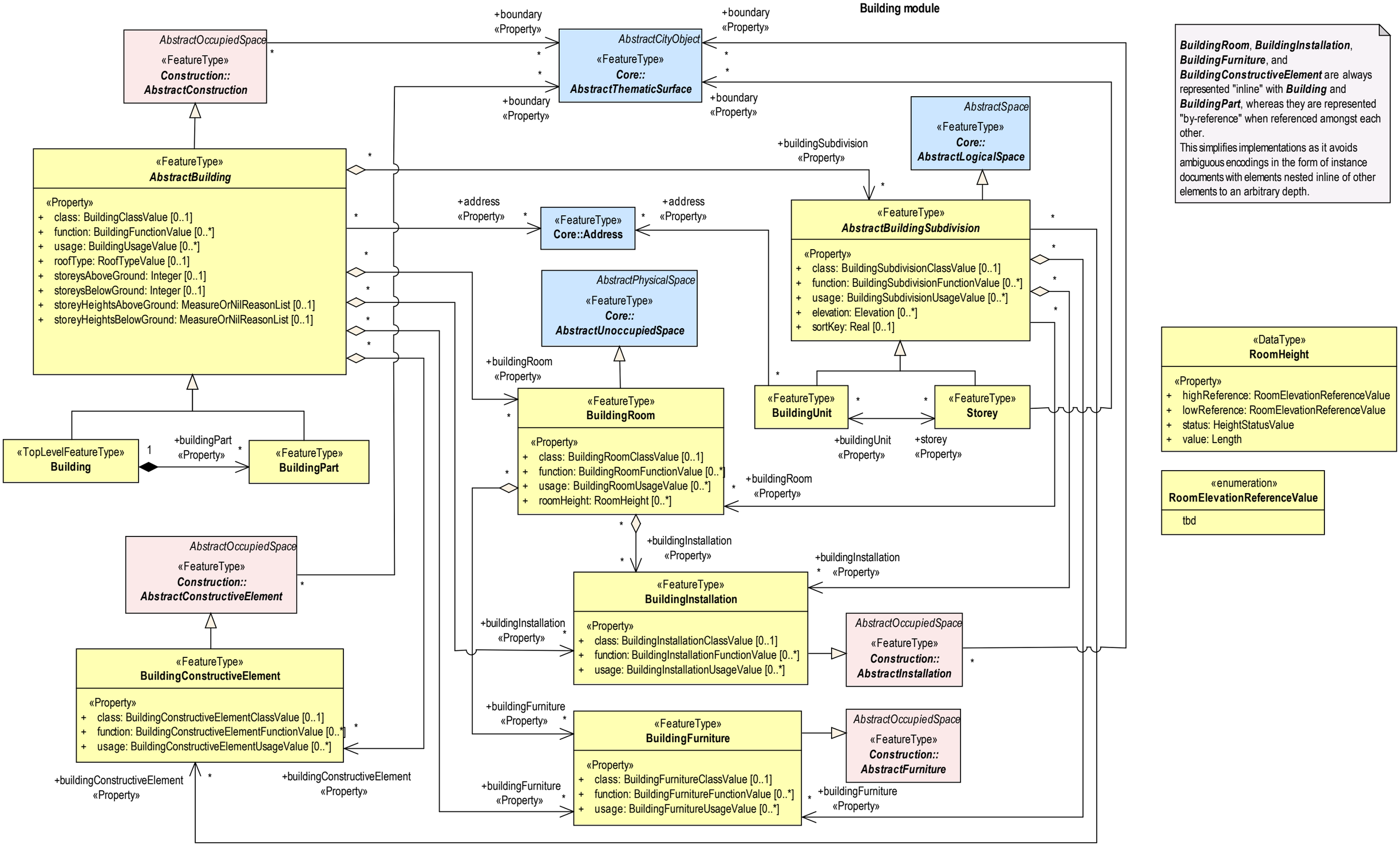
Bridge module



Bridge module - Code lists



Building module



BuildingRoom, ***BuildingInstallation***, ***BuildingFurniture***, and ***BuildingConstructiveElement*** are always represented "inline" with ***Building*** and ***BuildingPart***, whereas they are represented "by-reference" when referenced amongst each other.
This simplifies implementations as it avoids ambiguous encodings in the form of instance documents with elements nested inline of other elements to an arbitrary depth.

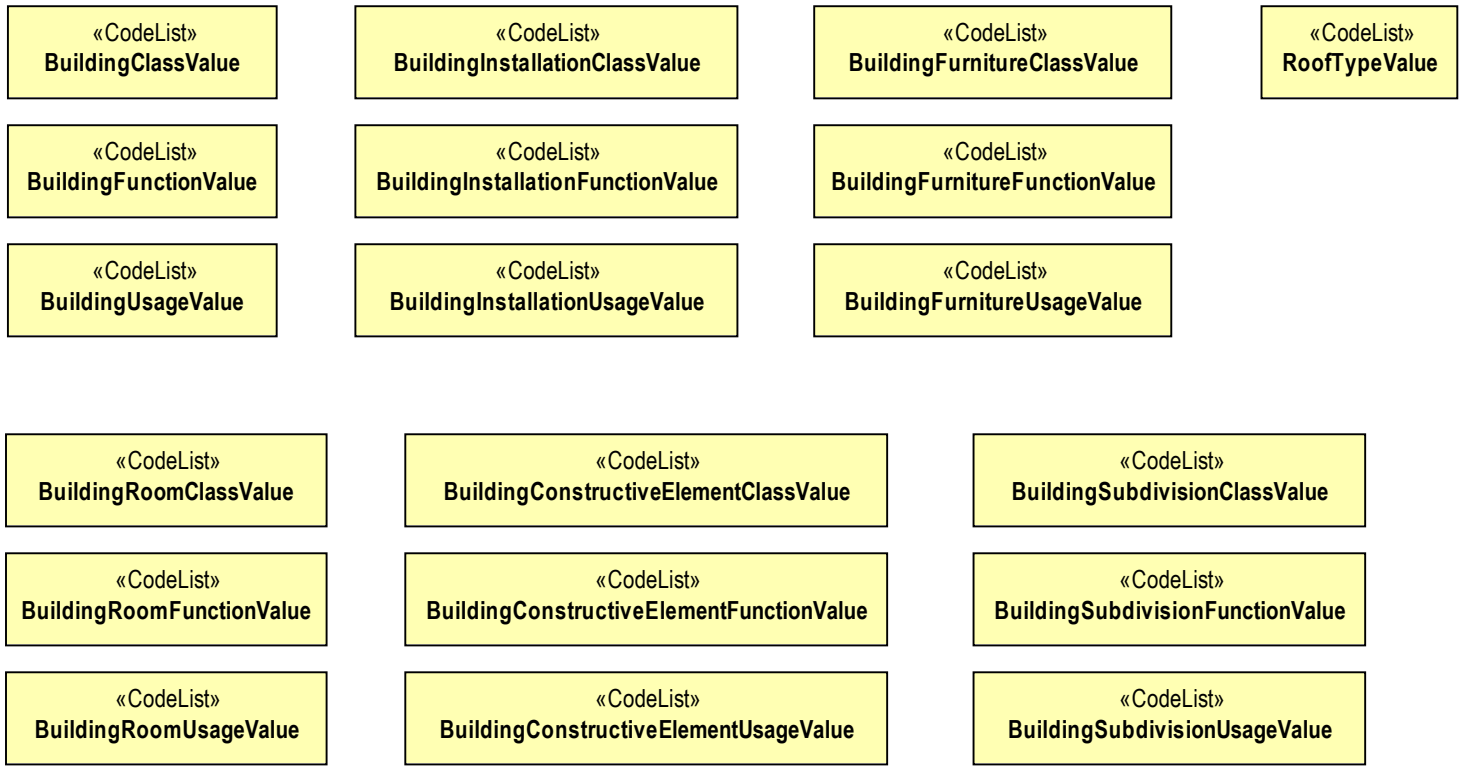
«DataType»
RoomHeight

«Property»
+ highReference: RoomElevationReferenceValue
+ lowReference: RoomElevationReferenceValue
+ status: HeightStatusValue
+ value: Length

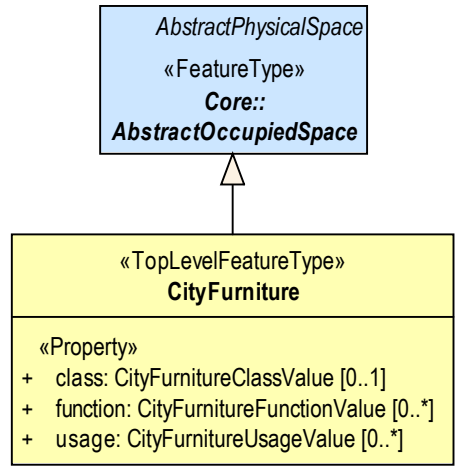
«enumeration»
RoomElevationReferenceValue

tbd

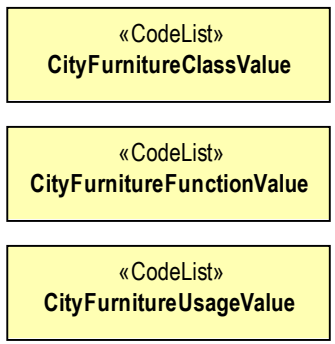
Building module - Code lists



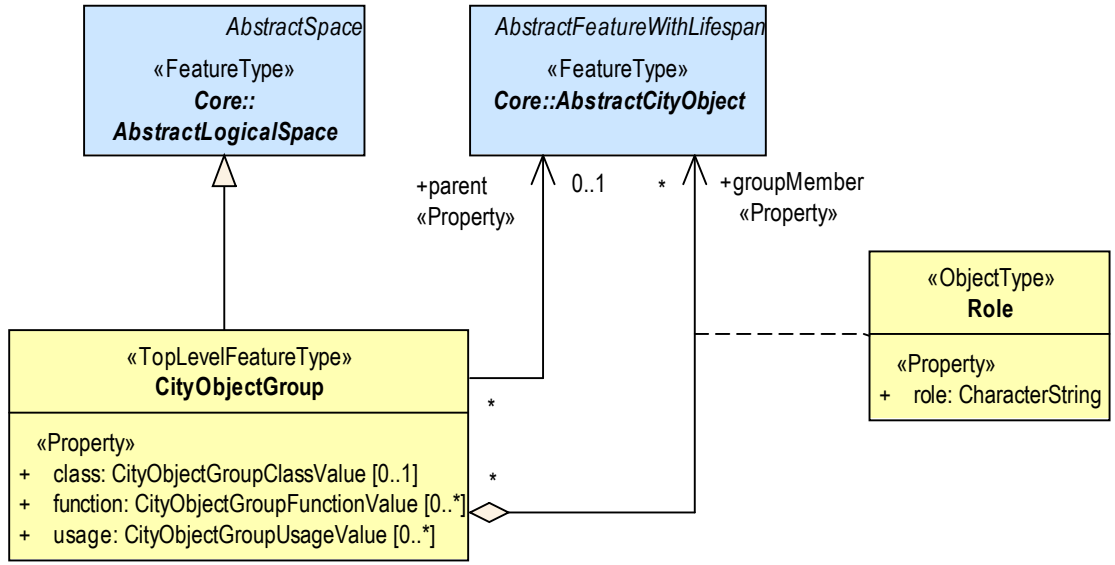
CityFurniture module



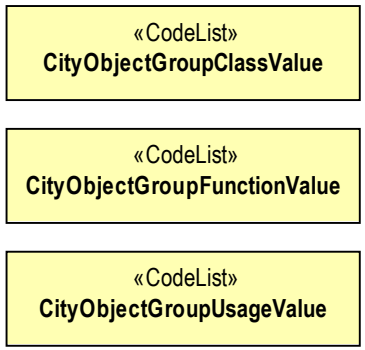
CityFurniture module - Code lists



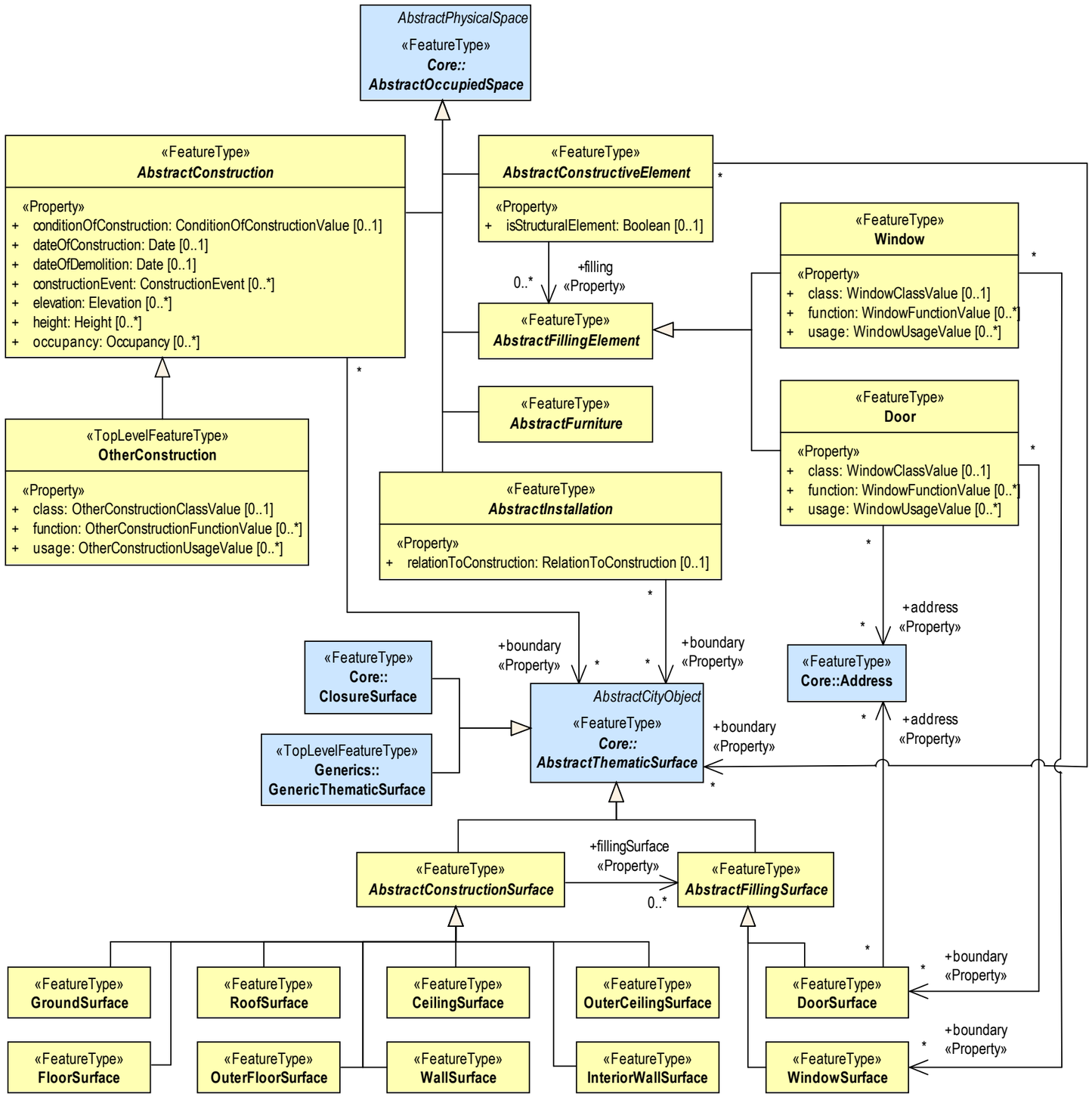
CityObjectGroup module



CityObjectGroup module - Code lists



Construction module



«enumeration»
ConditionOfConstructionValue

- declined
- demolished
- functional
- projected
- ruin
- underConstruction

«enumeration»
RelationToConstruction

- inside
- outside
- bothInsideAndOutside

«DataType»
Elevation

«Property»
+ elevationReference: ElevationReferenceValue
+ elevationValue: DirectPosition

«DataType»
Height

«Property»
+ highReference: ElevationReferenceValue
+ lowReference: ElevationReferenceValue
+ status: HeightStatusValue
+ value: Length

«enumeration»
ElevationReferenceValue

- aboveGroundEnvelope
- bottomOfConstruction
- entrancePoint
- generalEave
- generalGround
- generalRoof
- generalRoofEdge
- highestEave
- highestGroundPoint
- highestPoint
- highestRoofEdge
- lowestEave
- lowestFloorAboveGround
- lowestGroundPoint
- lowestRoofEdge
- topOfConstruction

«enumeration»
HeightStatusValue

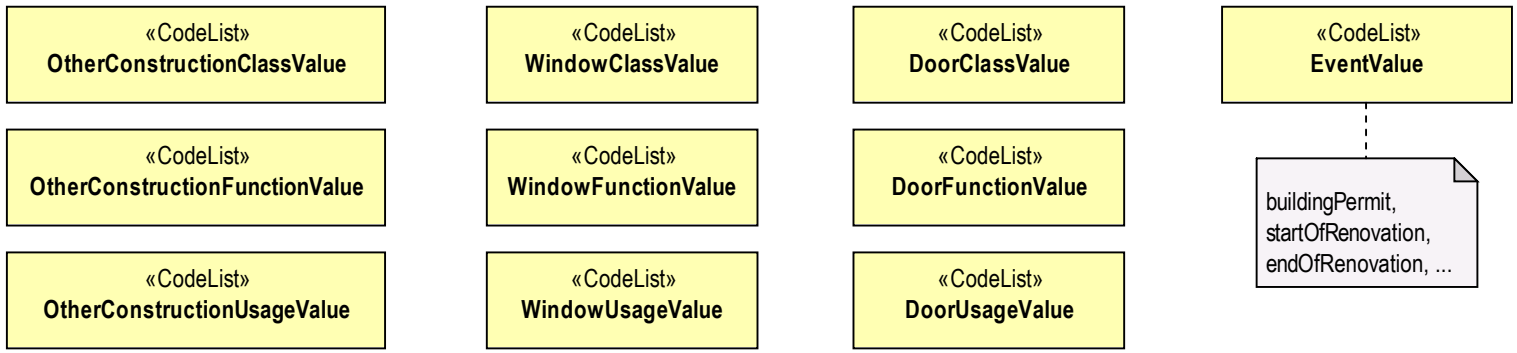
- estimated
- measured

Reference values for heights below ground will be added.

«DataType»
ConstructionEvent

«Property»
+ event: EventValue
+ dateOfEvent: Date
+ description: CharacterString [0..1]

Construction module - Code lists



Dynamizer module

AnyFeature
«FeatureType»
Core::
AbstractFeatureWithLifespan

«FeatureType»
Core::AbstractCityObject

+dynamizer
«Property»
*

«FeatureType»
Core::AbstractDynamizer

+sensorLocation
«Property»
0..1
0..*

«DataType»
SensorConnection
«Property»
+ connectionType: SensorConnectionValue
+ observationProperty: CharacterString
+ uom: CharacterString [0..1]
+ sensorID: CharacterString [0..1]
+ sensorName: CharacterString [0..1]
+ observationID: CharacterString [0..1]
+ datastreamID: CharacterString [0..1]
+ baseURL: URI [0..1]
+ authType: AuthenticationValue [0..1]
+ mqttServer: CharacterString [0..1]
+ mqttTopic: CharacterString [0..1]
+ linkToObservation: URI [0..1]
+ linkToSensorDescription: URI [0..1]

+sensorConnection
«Property»
0..1

«FeatureType»
Dynamizer
«Property»
+ attributeRef: URI
+ startTime: TM_Position [0..1]
+ endTime: TM_Position [0..1]

+dynamicData
«Property»
*
0..1

«FeatureType»
AbstractTimeseries
«Property»
+ firstTimestamp: TM_Position [0..1]
+ lastTimestamp: TM_Position [0..1]

+timeseries
«Property»
1
0..1

«DataType»
TimeseriesComponent
«Property»
+ repetitions: Integer
+ additionalGap: TM_Duration [0..1]

+isComponent
«Property»
1..*
{ordered}

«FeatureType»
AbstractAtomicTimeseries
«Property»
+ observationProperty: CharacterString
+ uom: CharacterString [0..1]

«FeatureType»
CompositeTimeseries

«FeatureType»
TabulatedFileTimeseries
«Property»
+ fileLocation: URI
+ fileType: TabulatedFileValue
+ mimeType: MimeValue [0..1]
+ valueType: TimeseriesValue
+ numberOfHeaderLines: Integer [0..1]
+ fieldSeparator: CharacterString
+ decimalSymbol: Character [0..1]
+ idColumnNo: Integer [0..1]
+ idColumnName: CharacterString [0..1]
+ idValue: CharacterString [0..1]
+ timeColumnNo: Integer [0..1]
+ timeColumnName: CharacterString [0..1]
+ valueColumnNo: Integer [0..1]
+ valueColumnName: CharacterString [0..1]

«FeatureType»
StandardFileTimeseries
«Property»
+ fileLocation: URI
+ fileType: StandardFileValue
+ mimeType: MimeValue [0..1]

«FeatureType»
GenericTimeseries
«Property»
+ valueType: TimeseriesValue

+timeValuePair
«Property»
1..*

«DataType»
TimeValuePair
«Property»
+ timestamp: TM_Position
+ intValue: Integer [0..1]
+ doubleValue: Real [0..1]
+ stringValue: CharacterString [0..1]
+ geometryValue: GM_Object [0..1]
+ uriValue: URI [0..1]
+ boolValue: Boolean [0..1]
+ implicitGeometryValue: ImplicitGeometry [0..1]
+ appearanceValue: AbstractAppearance [0..1]

«enumeration»
TimeseriesValue
integer
double
string
geometry
uri
bool
implicitGeometry
appearance

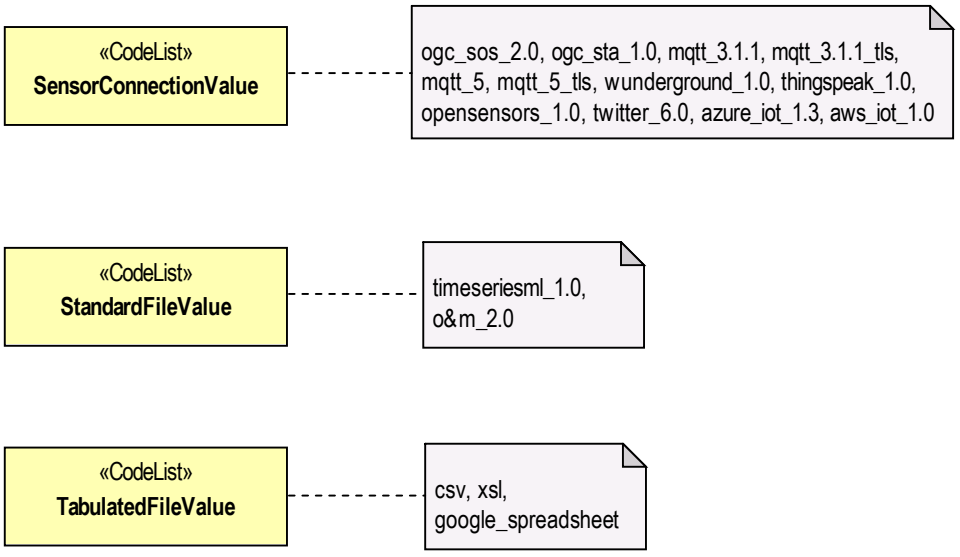
«enumeration»
AuthenticationValue
none
basic
oAuth2
apiKey
accessToken

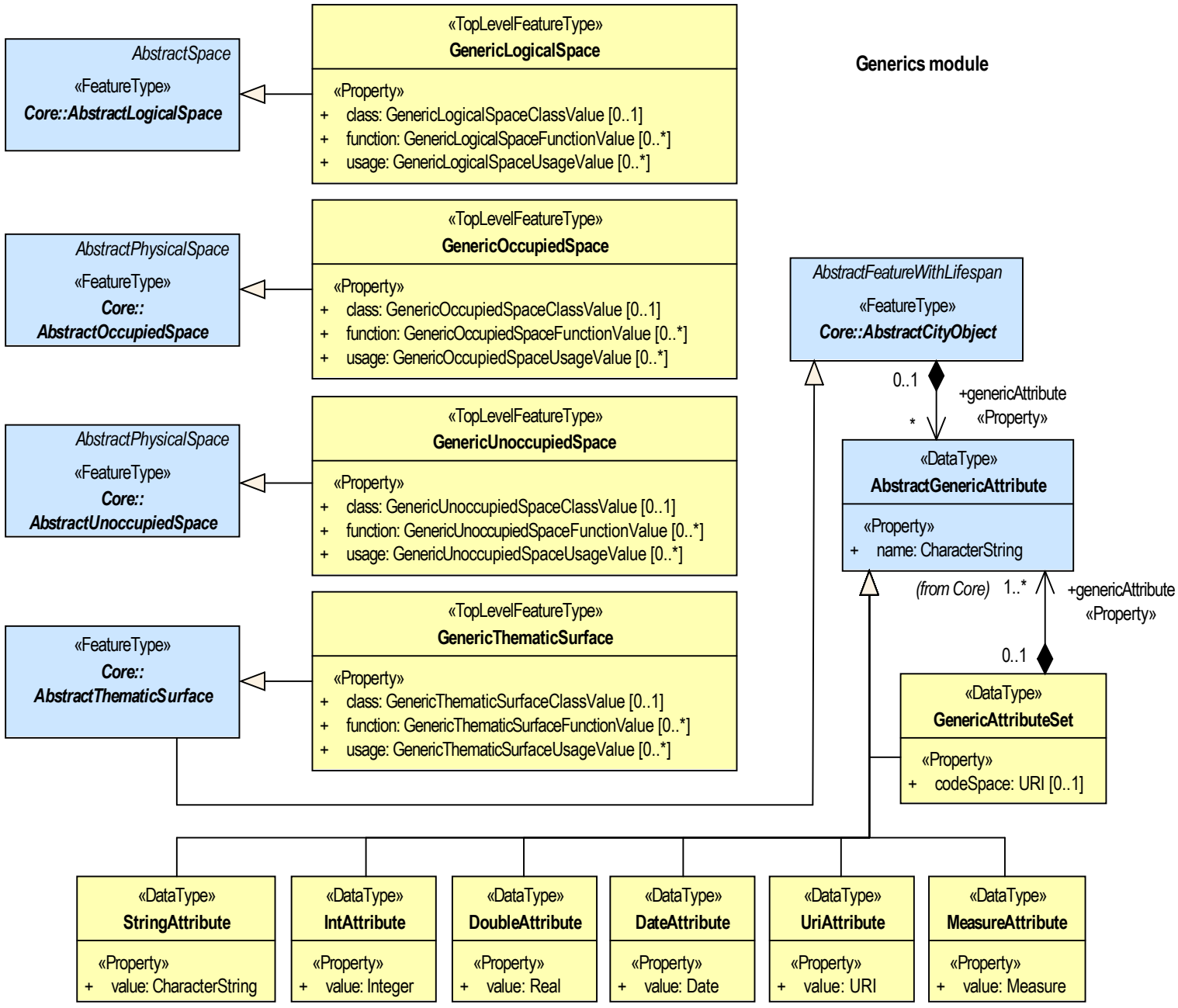
context TabulatedFileTimeseries **inv**:
(timeColumnNo->notEmpty() **or** timeColumnName->notEmpty()) **and**
(valueColumnNo->notEmpty() **or** valueColumnName->notEmpty()) **and**
(idValue->notEmpty() **implies** idColumnNo->notEmpty() **or** idColumnName->notEmpty())

context GenericTimeseries **inv**:
if valueType = TimeseriesValueType::integer **then**
TimeValuePair->forAll(c|c.intValue->size()=1)
else
if valueType = TimeseriesValueType::double **then**
TimeValuePair->forAll(c|c.doubleValue->size()=1)
else
if valueType = TimeseriesValueType::string **then**
TimeValuePair->forAll(c|c.stringValue->size()=1)
else
if valueType = TimeseriesValueType::geometry **then**
TimeValuePair->forAll(c|c.geometryValue->size()=1)
else
if valueType = TimeseriesValueType::uri **then**
TimeValuePair->forAll(c|c.uriValue->size()=1)
else
if valueType = TimeseriesValueType::bool **then**
TimeValuePair->forAll(c|c.boolValue->size()=1)
else
if valueType = TimeseriesValueType::implicitGeometry **then**
TimeValuePair->forAll(c|c.implicitGeometryValue->size()=1)
else
TimeValuePair->forAll(c|c.appearanceValue->size()=1)
endif
endif
endif
endif
endif
endif
endif

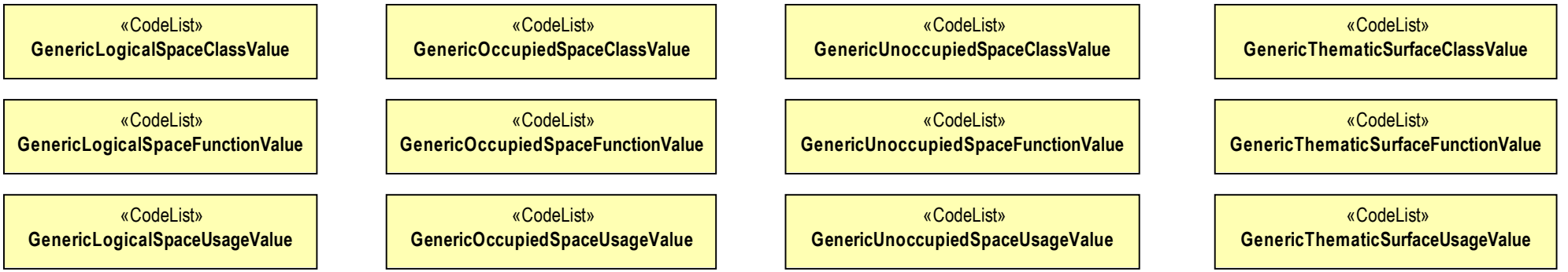
context TimeValuePair **inv**:
intValue->size() + doubleValue->size() + stringValue->size() +
geometryValue->size() + uriValue->size() + boolValue->size() +
implicitGeometryValue->size() + appearanceValue->size() = 1

Dynamizer module - Code lists

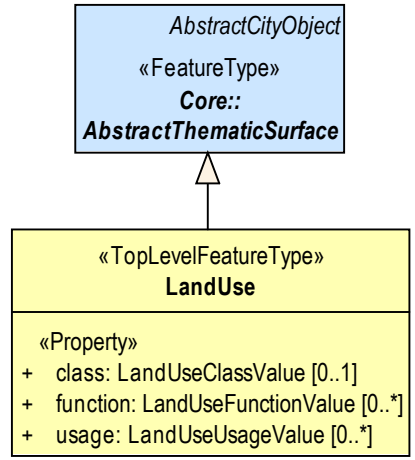



Generics module
AbstractSpace
«FeatureType»
Core::AbstractLogicalSpace
«TopLevelFeatureType»
GenericLogicalSpace
«Property»
+ class: GenericLogicalSpaceClassValue [0..1]
+ function: GenericLogicalSpaceFunctionValue [0..*]
+ usage: GenericLogicalSpaceUsageValue [0..*]
AbstractPhysicalSpace
«FeatureType»
Core::
AbstractOccupiedSpace
«TopLevelFeatureType»
GenericOccupiedSpace
«Property»
+ class: GenericOccupiedSpaceClassValue [0..1]
+ function: GenericOccupiedSpaceFunctionValue [0..*]
+ usage: GenericOccupiedSpaceUsageValue [0..*]
AbstractPhysicalSpace
«FeatureType»
Core::
AbstractUnoccupiedSpace
«TopLevelFeatureType»
GenericUnoccupiedSpace
«Property»
+ class: GenericUnoccupiedSpaceClassValue [0..1]
+ function: GenericUnoccupiedSpaceFunctionValue [0..*]
+ usage: GenericUnoccupiedSpaceUsageValue [0..*]
«FeatureType»
Core::
AbstractThematicSurface
«TopLevelFeatureType»
GenericThematicSurface
«Property»
+ class: GenericThematicSurfaceClassValue [0..1]
+ function: GenericThematicSurfaceFunctionValue [0..*]
+ usage: GenericThematicSurfaceUsageValue [0..*]
AbstractFeatureWithLifespan
«FeatureType»
Core::AbstractCityObject
0..1
*
+genericAttribute
«Property»
«DataType»
AbstractGenericAttribute
«Property»
+ name: CharacterString
(from Core)
1..*
+genericAttribute
«Property»
0..1
«DataType»
GenericAttributeSet
«Property»
+ codeSpace: URI [0..1]
«DataType»
StringAttribute
«Property»
+ value: CharacterString
«DataType»
IntAttribute
«Property»
+ value: Integer
«DataType»
DoubleAttribute
«Property»
+ value: Real
«DataType»
DateAttribute
«Property»
+ value: Date
«DataType»
UriAttribute
«Property»
+ value: URI
«DataType»
MeasureAttribute
«Property»
+ value: Measure

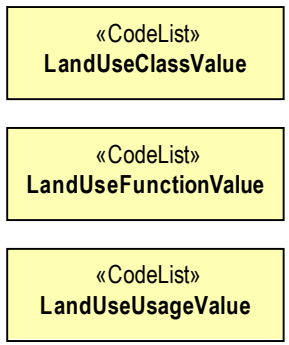
Generics module - Code lists



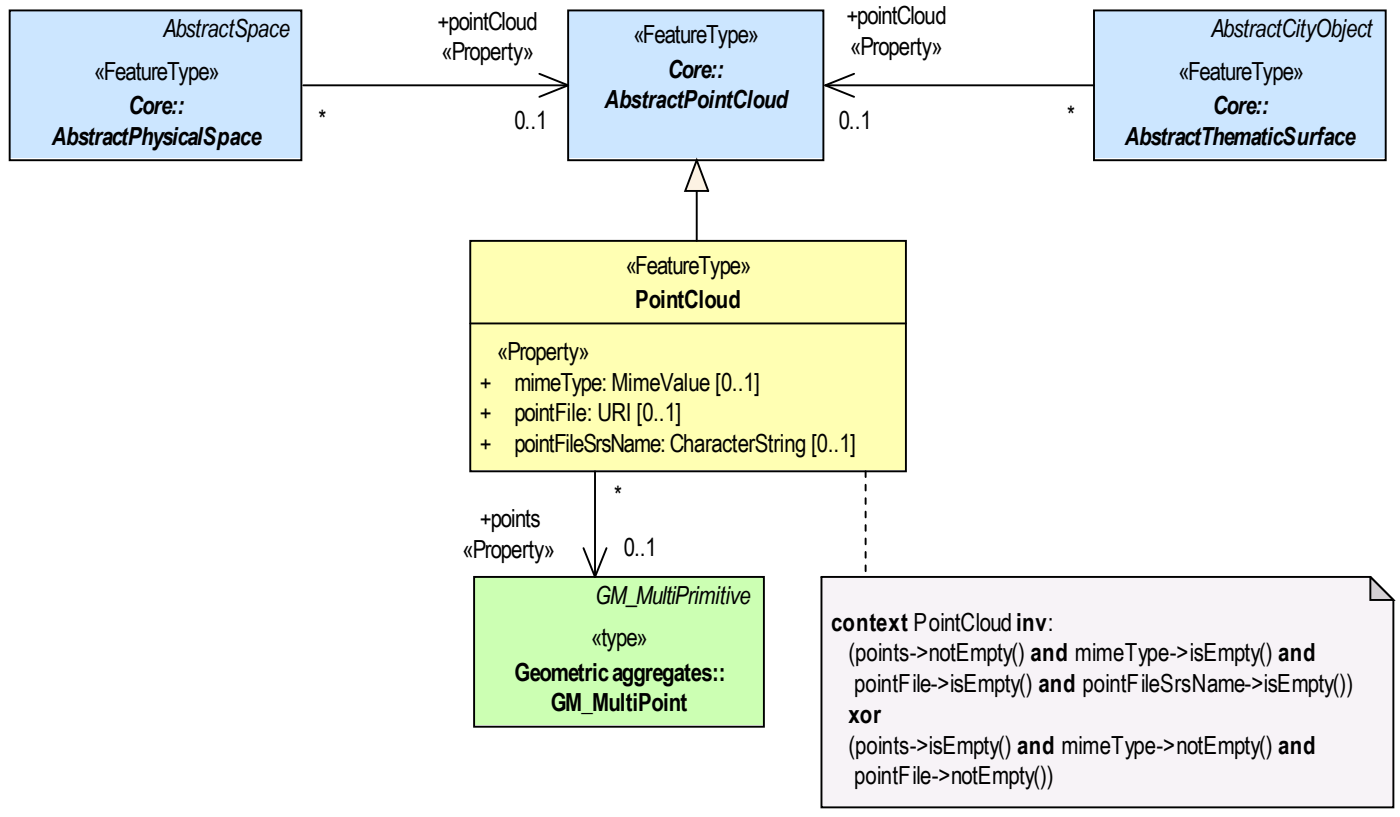
LandUse module

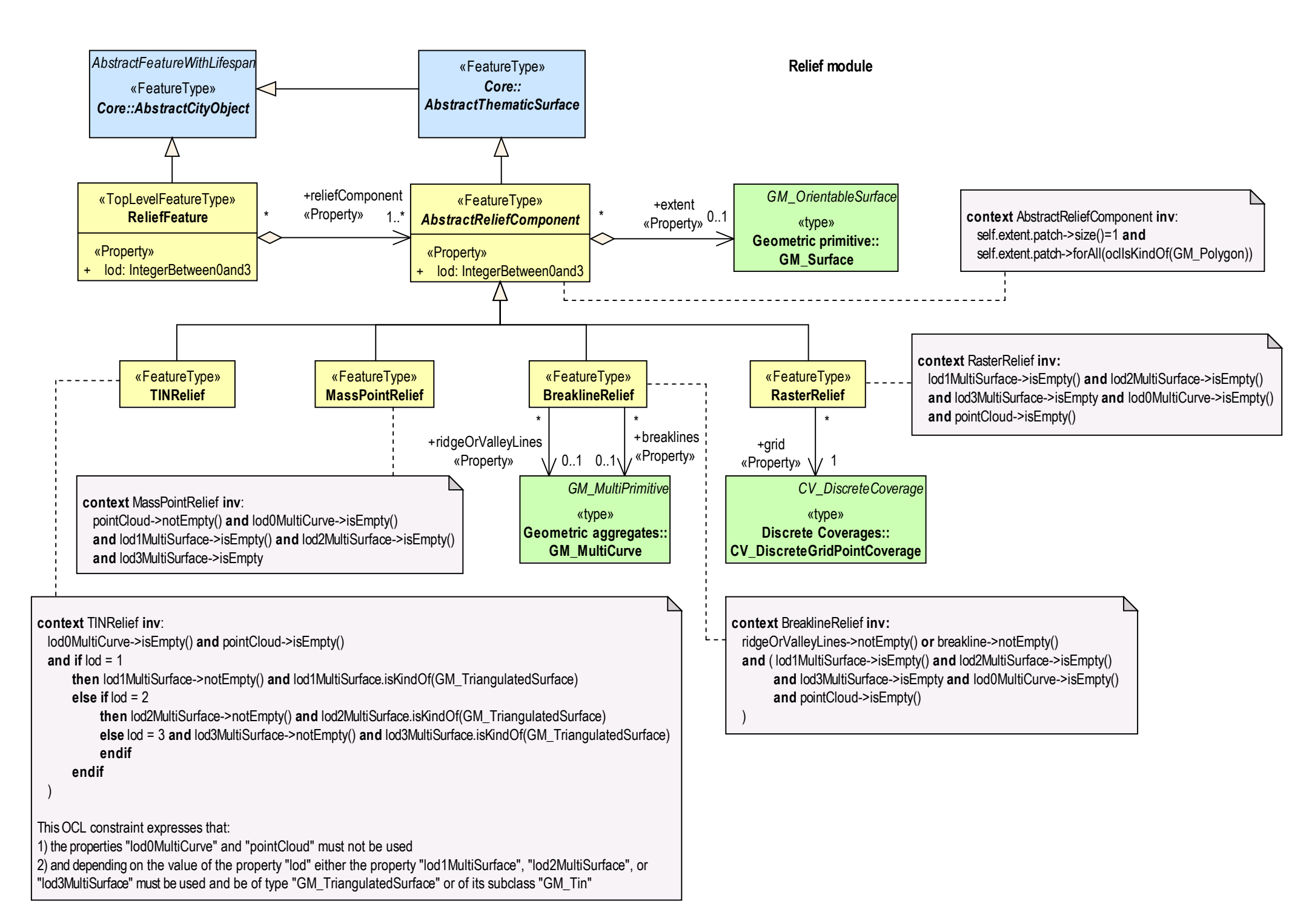


LandUse module - Code lists

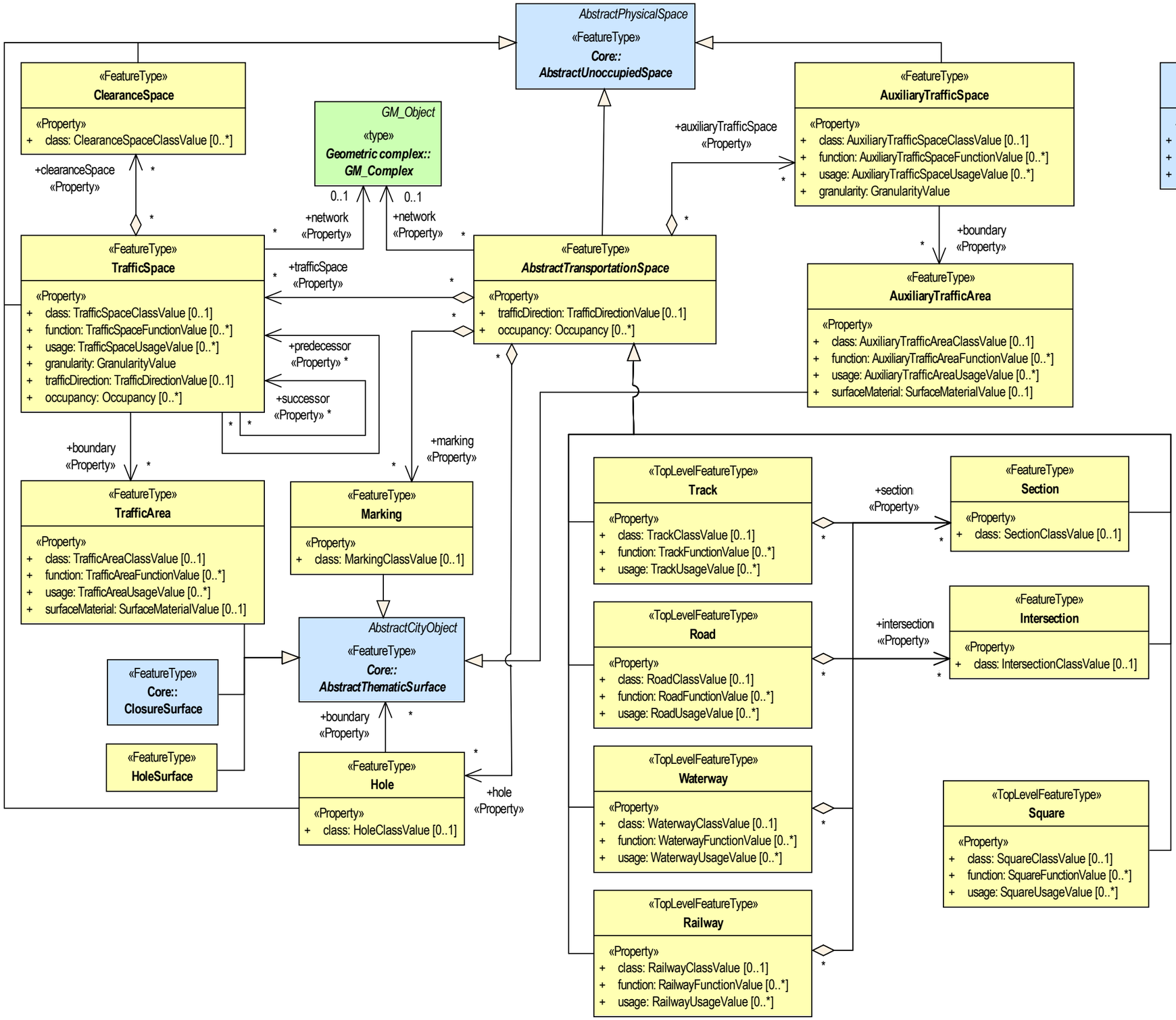


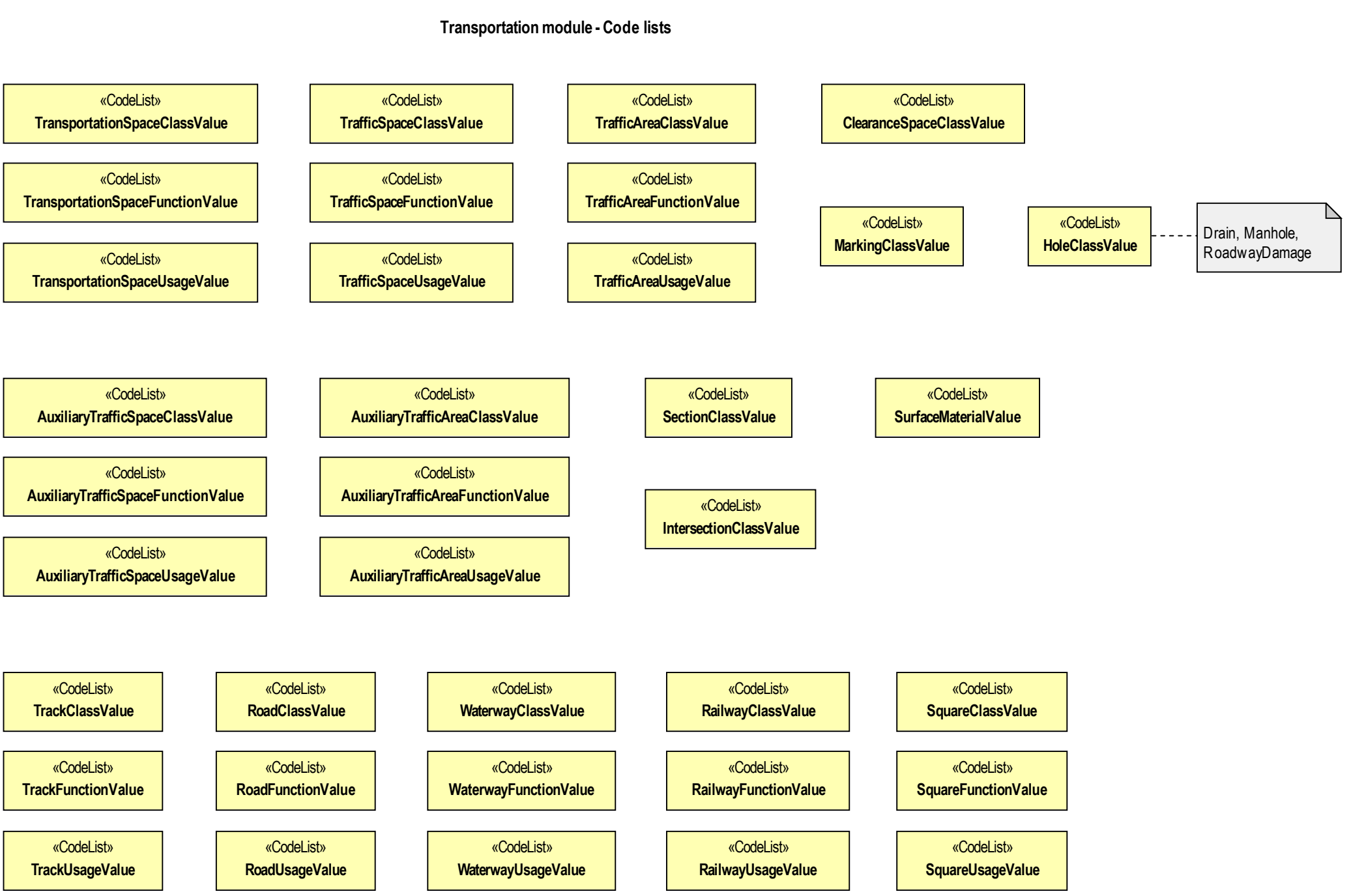
PointCloud module

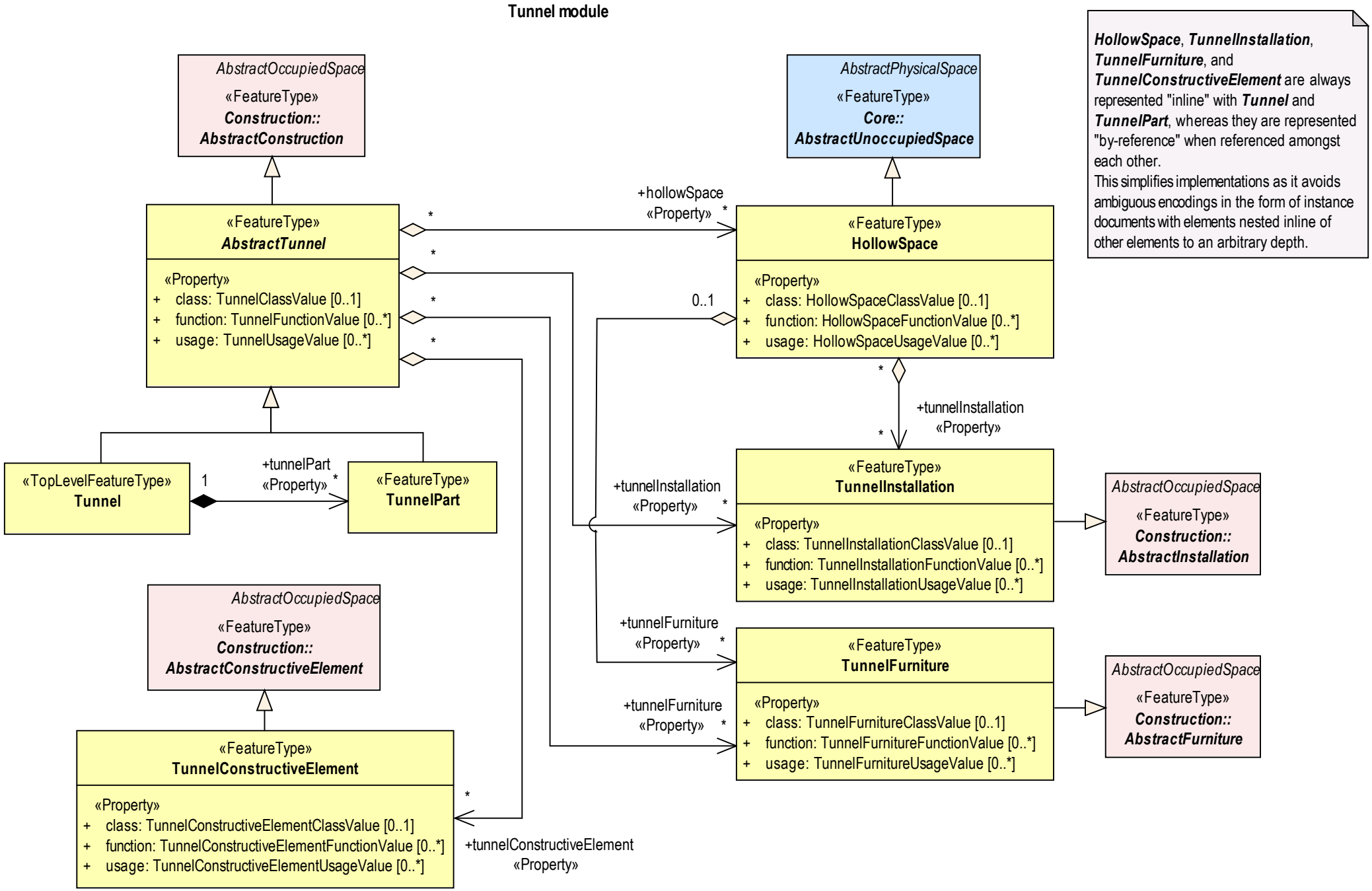



Relief module
AbstractFeatureWithLifespan
«FeatureType»
Core::AbstractCityObject
«FeatureType»
Core::
AbstractThematicSurface
«TopLevelFeatureType»
ReliefFeature
«Property»
+ lod: IntegerBetween0and3
+reliefComponent
«Property»
1..*
«FeatureType»
AbstractReliefComponent
«Property»
+ lod: IntegerBetween0and3
+extent
«Property»
0..1
GM_OrientableSurface
«type»
Geometric primitive::
GM_Surface
context AbstractReliefComponent inv:
self.extent.patch->size()=1 and
self.extent.patch->forAll(oclIsKindOf(GM_Polygon))
«FeatureType»
TINRelief
«FeatureType»
MassPointRelief
«FeatureType»
BreaklineRelief
«FeatureType»
RasterRelief
context RasterRelief inv:
lod1MultiSurface->isEmpty() and lod2MultiSurface->isEmpty()
and lod3MultiSurface->isEmpty and lod0MultiCurve->isEmpty()
and pointCloud->isEmpty()
+ridgeOrValleyLines
«Property»
0..1
0..1
+breaklines
«Property»
+grid
«Property»
1
GM_MultiPrimitive
«type»
Geometric aggregates::
GM_MultiCurve
CV_DiscreteCoverage
«type»
Discrete Coverages::
CV_DiscreteGridPointCoverage
context MassPointRelief inv:
pointCloud->notEmpty() and lod0MultiCurve->isEmpty()
and lod1MultiSurface->isEmpty() and lod2MultiSurface->isEmpty()
and lod3MultiSurface->isEmpty
context TINRelief inv:
lod0MultiCurve->isEmpty() and pointCloud->isEmpty()
and if lod = 1
then lod1MultiSurface->notEmpty() and lod1MultiSurface.isKindOf(GM_TriangulatedSurface)
else if lod = 2
then lod2MultiSurface->notEmpty() and lod2MultiSurface.isKindOf(GM_TriangulatedSurface)
else lod = 3 and lod3MultiSurface->notEmpty() and lod3MultiSurface.isKindOf(GM_TriangulatedSurface)
endif
endif
)
This OCL constraint expresses that:
1) the properties "lod0MultiCurve" and "pointCloud" must not be used
2) and depending on the value of the property "lod" either the property "lod1MultiSurface", "lod2MultiSurface", or
"lod3MultiSurface" must be used and be of type "GM_TriangulatedSurface" or of its subclass "GM_Tin"
context BreaklineRelief inv:
ridgeOrValleyLines->notEmpty() or breakline->notEmpty()
and (lod1MultiSurface->isEmpty() and lod2MultiSurface->isEmpty()
and lod3MultiSurface->isEmpty and lod0MultiCurve->isEmpty()
and pointCloud->isEmpty()
)

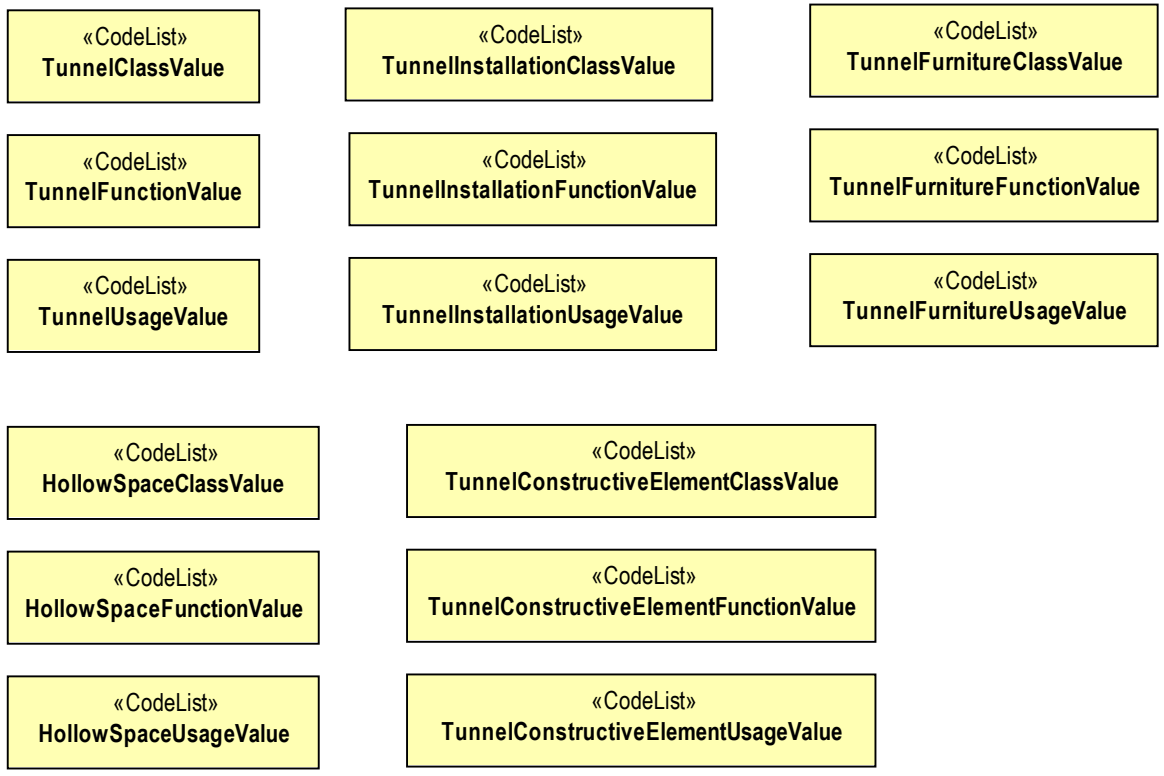
Transportation module



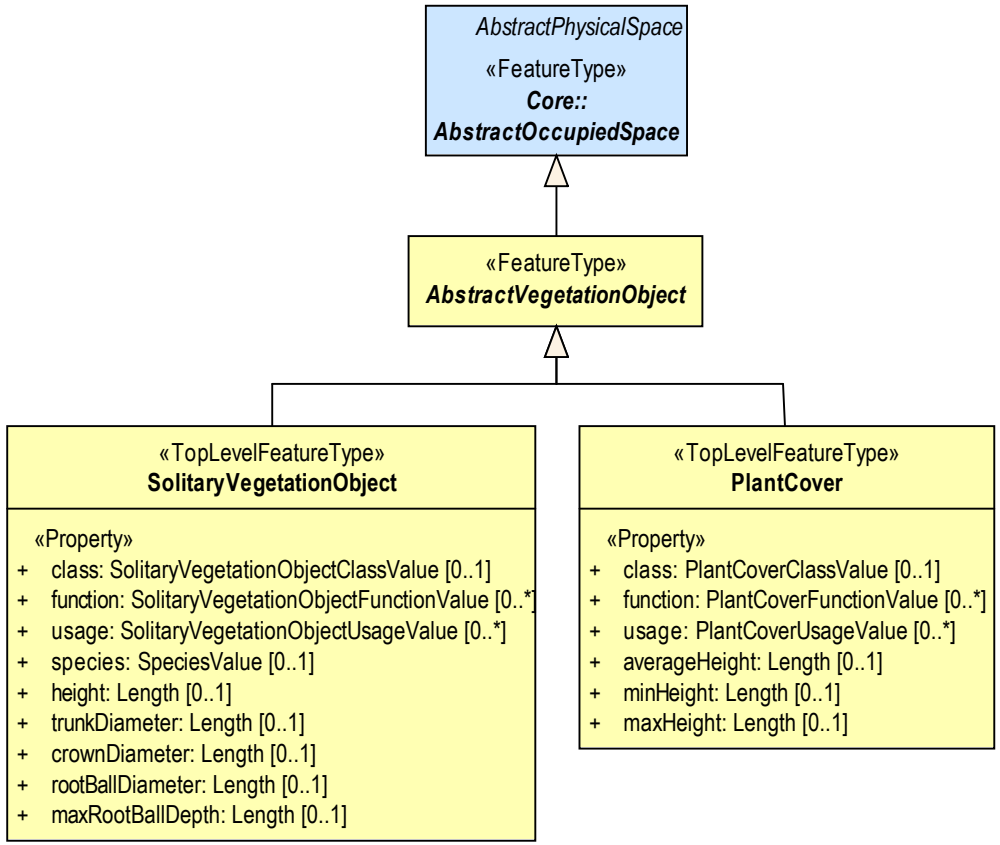

Transportation module - Code lists
«CodeList»
TransportationSpaceClassValue
«CodeList»
TrafficSpaceClassValue
«CodeList»
TrafficAreaClassValue
«CodeList»
ClearanceSpaceClassValue
«CodeList»
TransportationSpaceFunctionValue
«CodeList»
TrafficSpaceFunctionValue
«CodeList»
TrafficAreaFunctionValue
«CodeList»
MarkingClassValue
«CodeList»
HoleClassValue
Drain, Manhole, RoadwayDamage
«CodeList»
TransportationSpaceUsageValue
«CodeList»
TrafficSpaceUsageValue
«CodeList»
TrafficAreaUsageValue
«CodeList»
AuxiliaryTrafficSpaceClassValue
«CodeList»
AuxiliaryTrafficAreaClassValue
«CodeList»
SectionClassValue
«CodeList»
SurfaceMaterialValue
«CodeList»
AuxiliaryTrafficSpaceFunctionValue
«CodeList»
AuxiliaryTrafficAreaFunctionValue
«CodeList»
IntersectionClassValue
«CodeList»
AuxiliaryTrafficSpaceUsageValue
«CodeList»
AuxiliaryTrafficAreaUsageValue
«CodeList»
TrackClassValue
«CodeList»
RoadClassValue
«CodeList»
WaterwayClassValue
«CodeList»
RailwayClassValue
«CodeList»
SquareClassValue
«CodeList»
TrackFunctionValue
«CodeList»
RoadFunctionValue
«CodeList»
WaterwayFunctionValue
«CodeList»
RailwayFunctionValue
«CodeList»
SquareFunctionValue
«CodeList»
TrackUsageValue
«CodeList»
RoadUsageValue
«CodeList»
WaterwayUsageValue
«CodeList»
RailwayUsageValue
«CodeList»
SquareUsageValue


Tunnel module
AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractConstruction
AbstractPhysicalSpace
«FeatureType»
Core::
AbstractUnoccupiedSpace
HollowSpace, TunnelInstallation, TunnelFurniture, and TunnelConstructiveElement are always represented "inline" with Tunnel and TunnelPart, whereas they are represented "by-reference" when referenced amongst each other.
This simplifies implementations as it avoids ambiguous encodings in the form of instance documents with elements nested inline of other elements to an arbitrary depth.
«FeatureType»
AbstractTunnel
«Property»
+ class: TunnelClassValue [0..1]
+ function: TunnelFunctionValue [0..*]
+ usage: TunnelUsageValue [0..*]
+hollowSpace
«Property»
«FeatureType»
HollowSpace
«Property»
+ class: HollowSpaceClassValue [0..1]
+ function: HollowSpaceFunctionValue [0..*]
+ usage: HollowSpaceUsageValue [0..*]
0..1
+tunnelInstallation
«Property»
«TopLevelFeatureType»
Tunnel
1
+tunnelPart
«Property»
«FeatureType»
TunnelPart
+tunnelInstallation
«Property»
«FeatureType»
TunnelInstallation
«Property»
+ class: TunnelInstallationClassValue [0..1]
+ function: TunnelInstallationFunctionValue [0..*]
+ usage: TunnelInstallationUsageValue [0..*]
AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractInstallation
AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractConstructiveElement
+tunnelFurniture
«Property»
«FeatureType»
TunnelFurniture
«Property»
+ class: TunnelFurnitureClassValue [0..1]
+ function: TunnelFurnitureFunctionValue [0..*]
+ usage: TunnelFurnitureUsageValue [0..*]
AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractFurniture
«FeatureType»
TunnelConstructiveElement
«Property»
+ class: TunnelConstructiveElementClassValue [0..1]
+ function: TunnelConstructiveElementFunctionValue [0..*]
+ usage: TunnelConstructiveElementUsageValue [0..*]
+tunnelConstructiveElement
«Property»

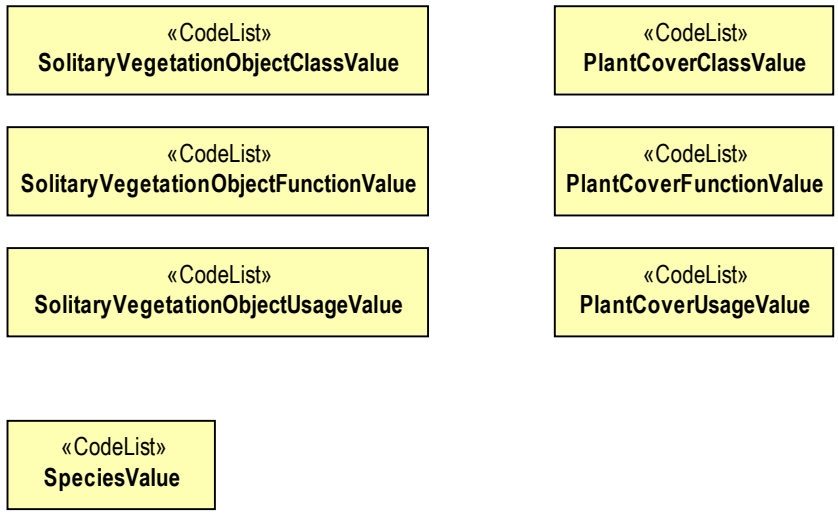
Tunnel module - Code lists



Vegetation module

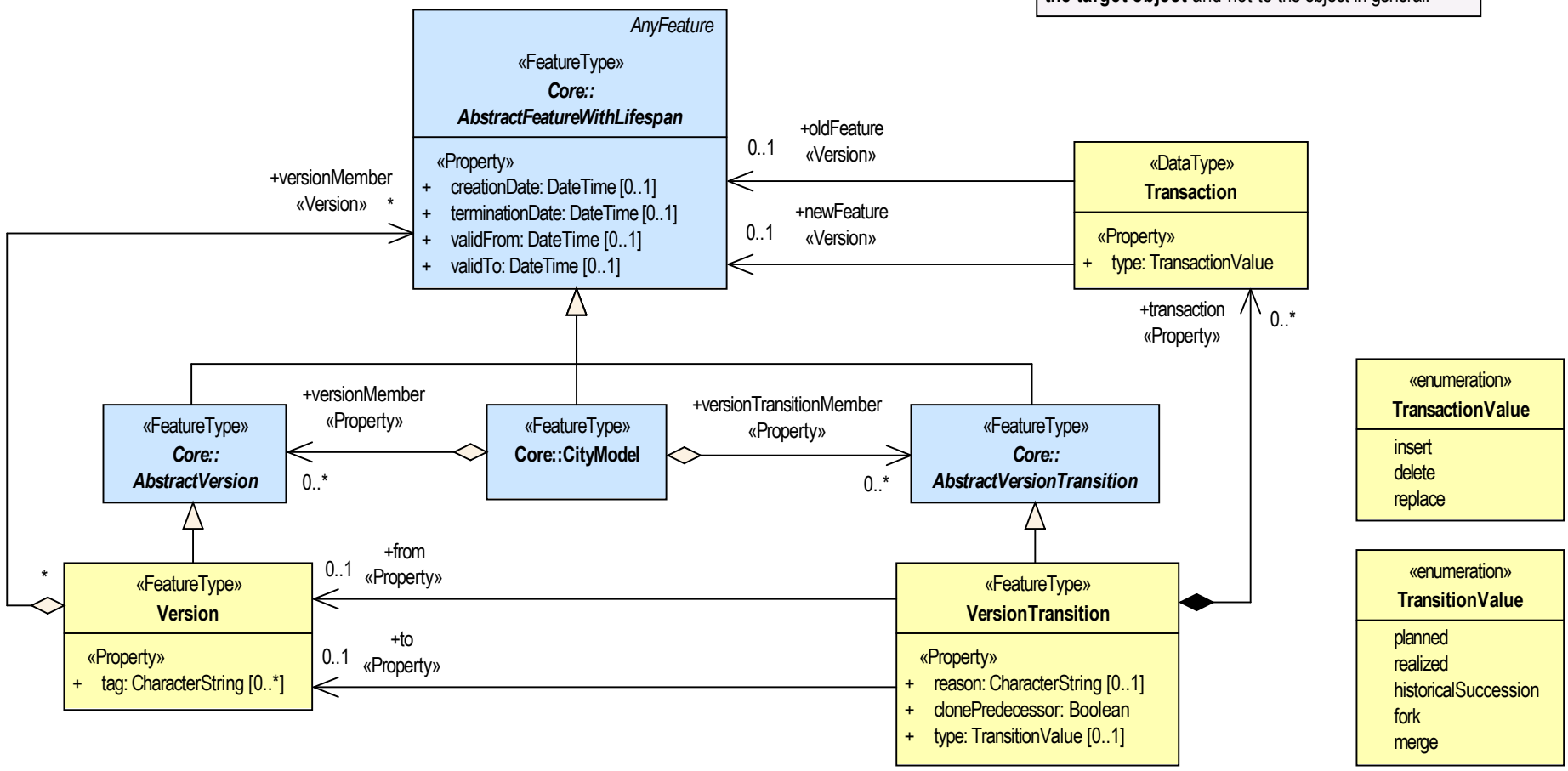


Vegetation module - Code lists

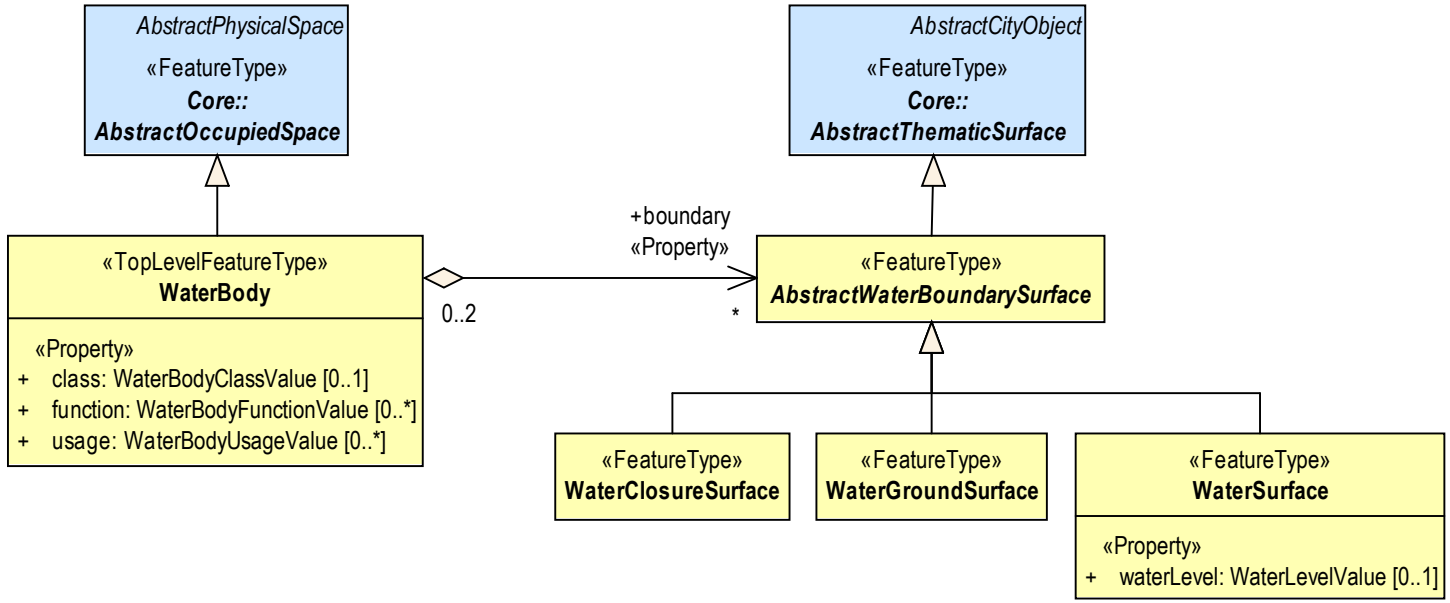


Versioning module

The **stereotype «Version»** is adopted from INSPIRE. The stereotype is used for association roles to express that the **association refers to a specific version of the target object** and not to the object in general.



WaterBody module



WaterBody module - Code lists