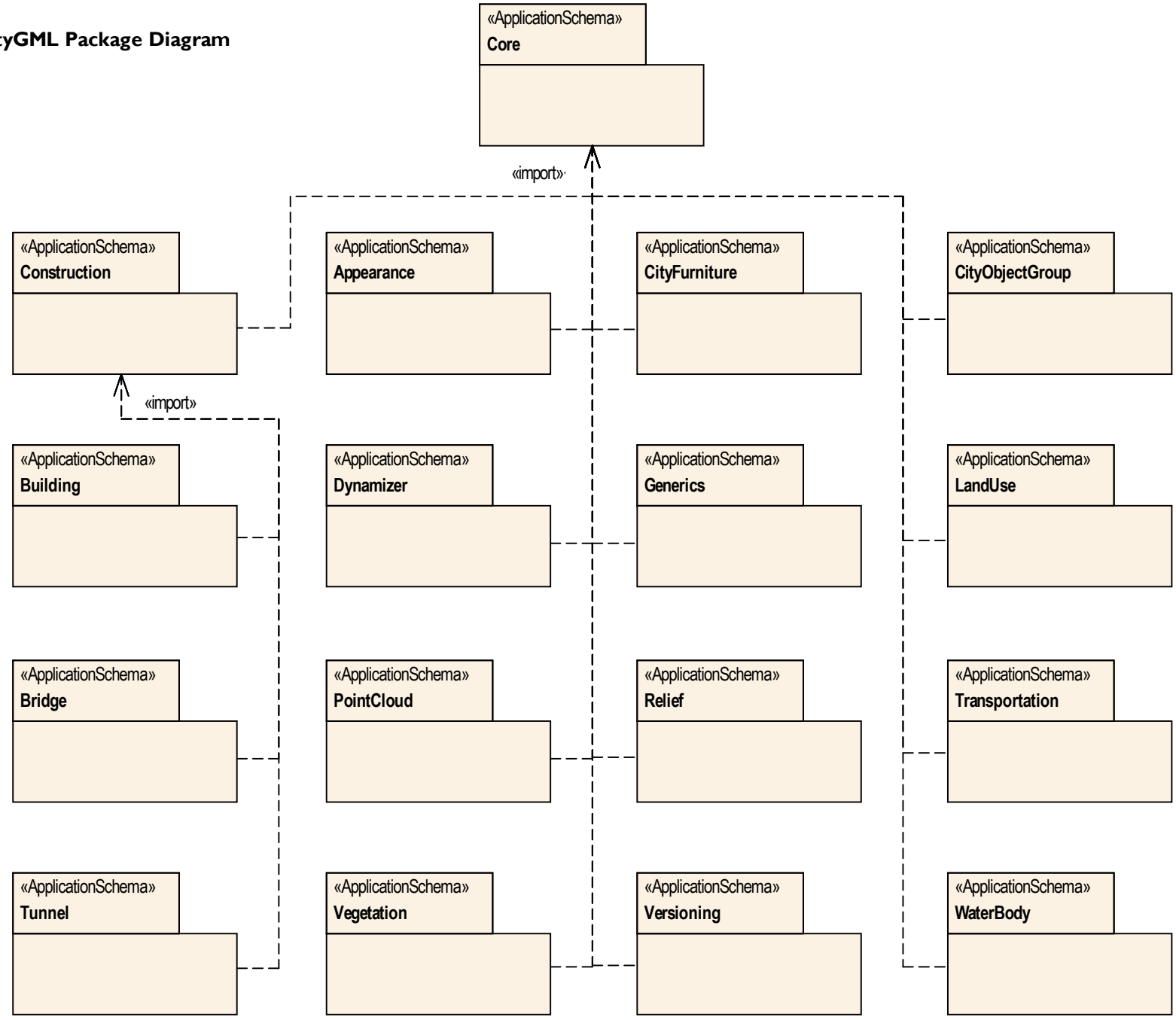
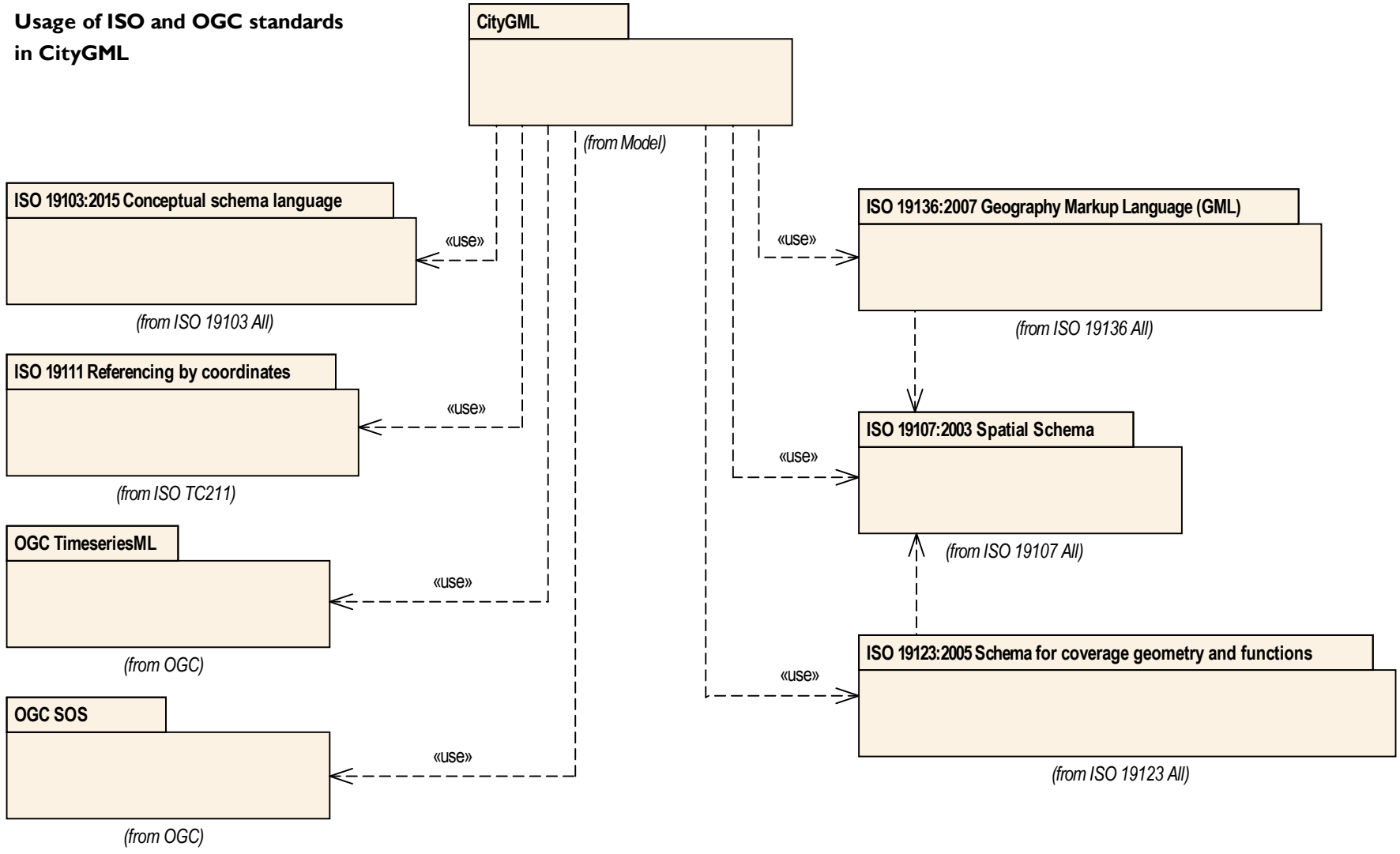
CityGML Package Diagram



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

«FeatureType» **CityModel** — * +cityObjectMember «Property» *

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

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

CityObjectRelation
«Property»
+ relationType: RelationTypeValue

+relatedTo «Property» * — +generalizesTo «Property» *

GM_Primitive «type» **Geometric primitive:: GM_Solid** — 0..1 +lod1Solid «Property» * / +lod2Solid «Property» * / +lod3Solid «Property» *
GM_Primitive «type» **Geometric primitive:: GM_Point** — 0..1 +lod0Point «Property» *

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

* +lod0MultiSurface «Property» / * +lod2MultiSurface «Property» / * +lod3MultiSurface «Property» — 0..1 — *GM_MultiPrimitive* «type» **Geometric aggregates:: GM_MultiSurface**
0..1 +lod1MultiSurface «Property» * / +lod2MultiSurface «Property» * / +lod3MultiSurface «Property» *

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

* +bounds «Property» — +boundary «Property» *

* +lod2MultiCurve «Property» / * +lod3MultiCurve «Property» — 0..1 — *GM_MultiPrimitive* «type» **Geometric aggregates:: GM_MultiCurve** — 0..1 +lod0MultiCurve «Property» *

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

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

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

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

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

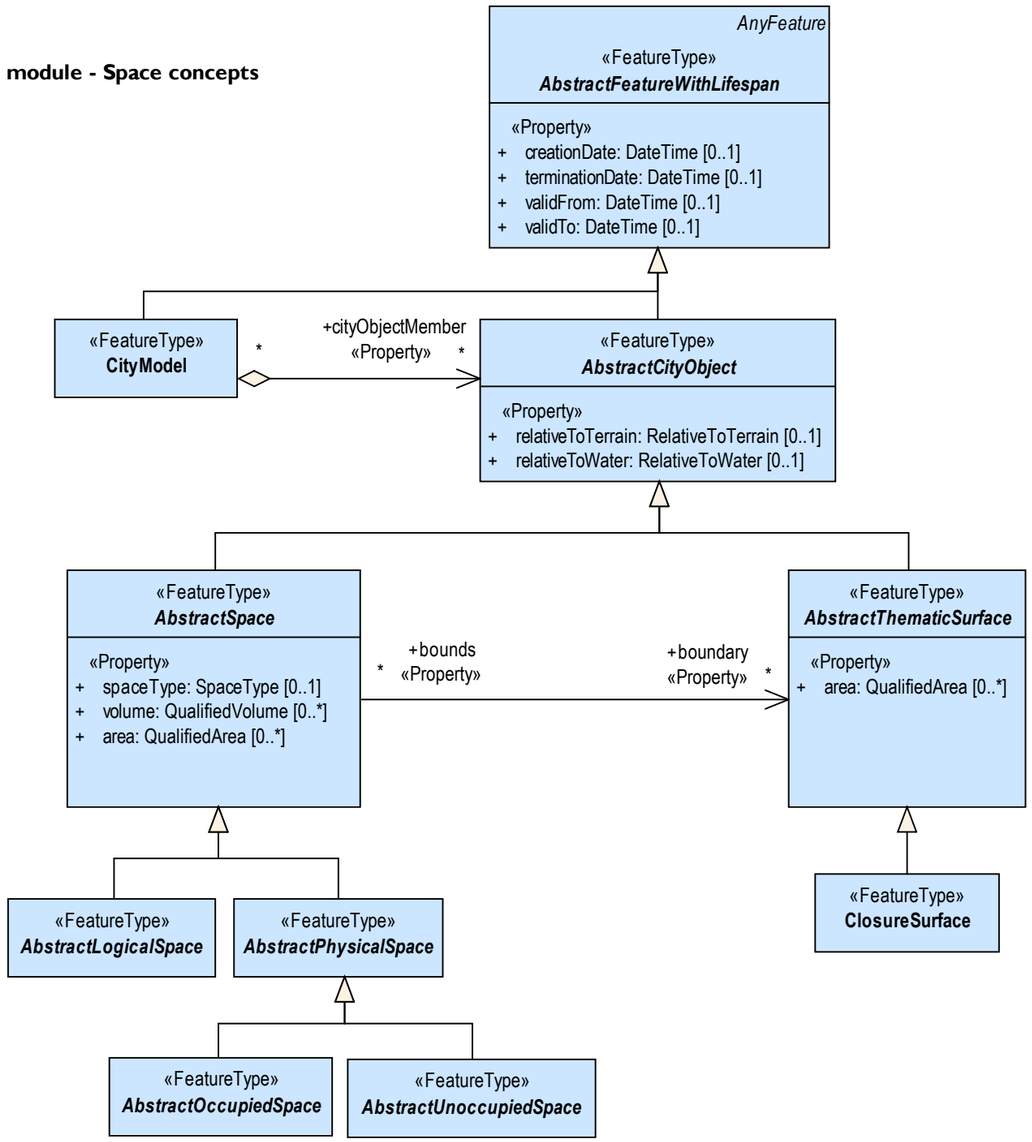
+lod1TerrainIntersectionCurve «Property» / +lod2TerrainIntersectionCurve «Property» / +lod3TerrainIntersectionCurve «Property» — * — 0..1

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

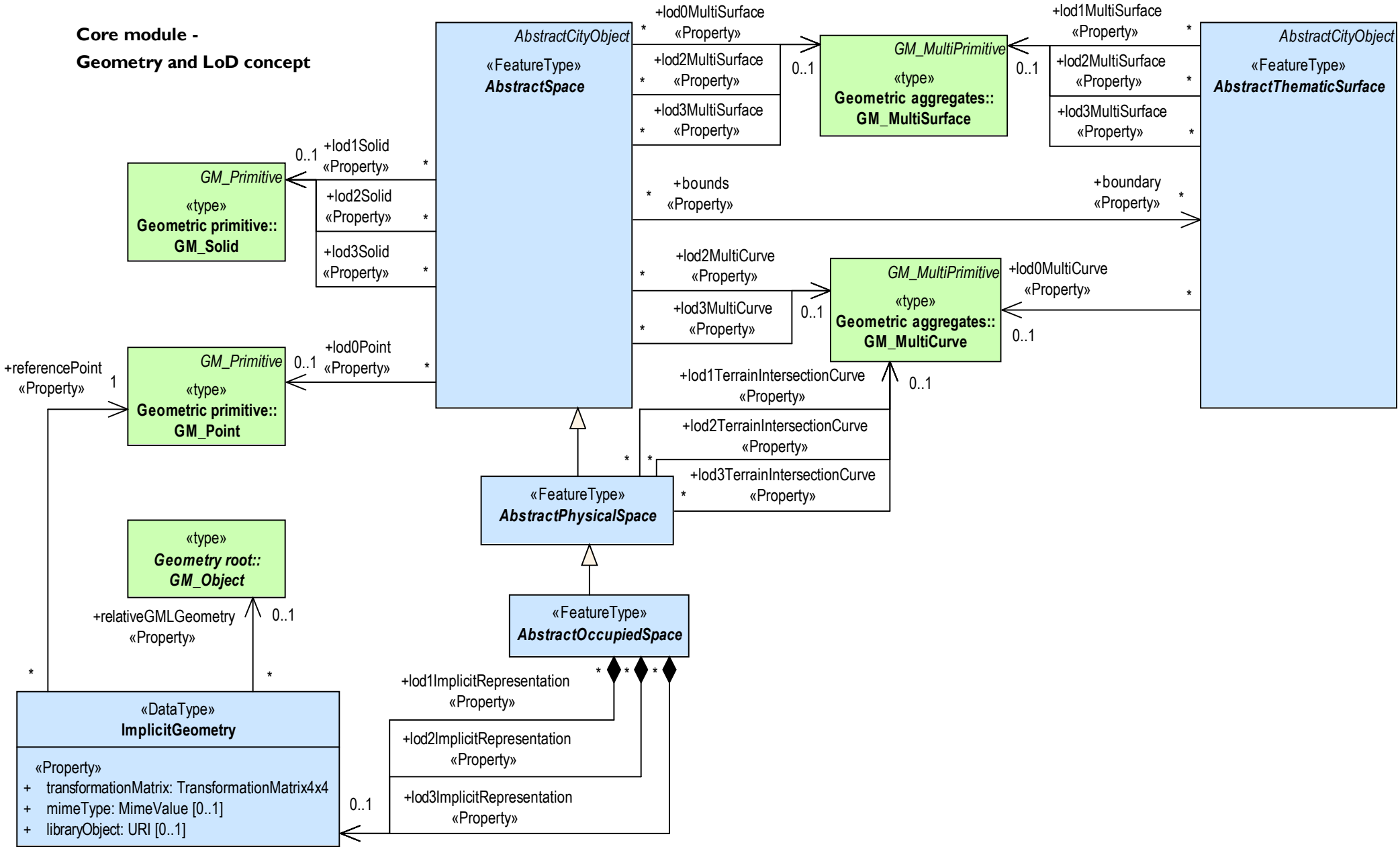
«FeatureType» ***AbstractOccupiedSpace***
«FeatureType» ***AbstractUnoccupiedSpace***

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

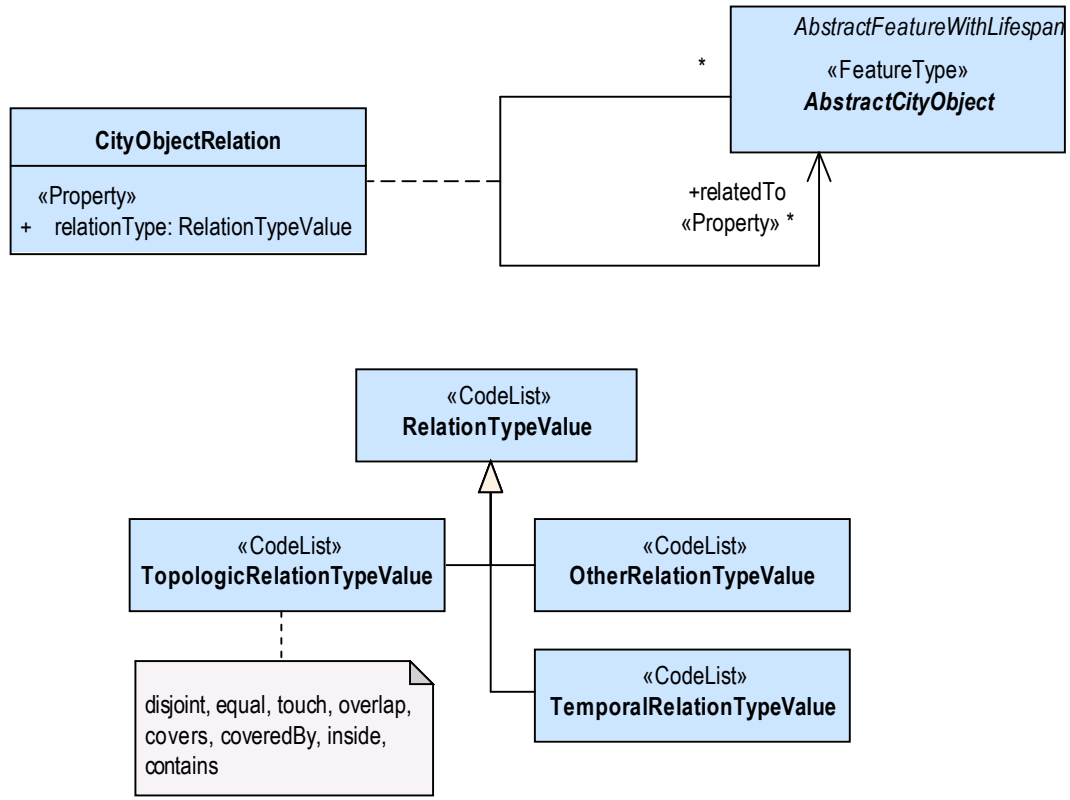
Core module - Space concepts



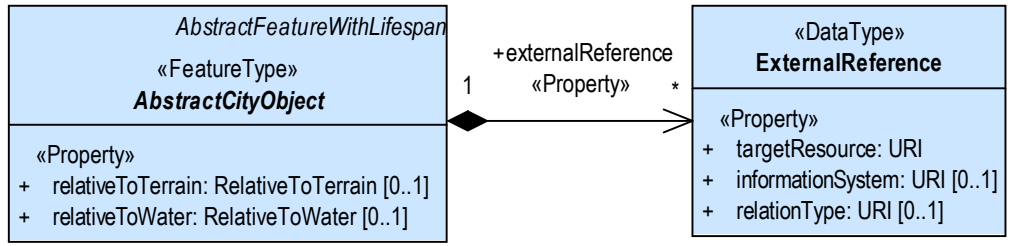
Core module -
Geometry and LoD concept



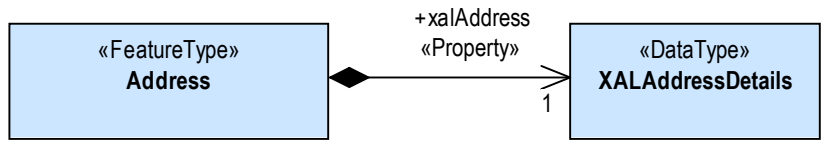
Core module - City object relations



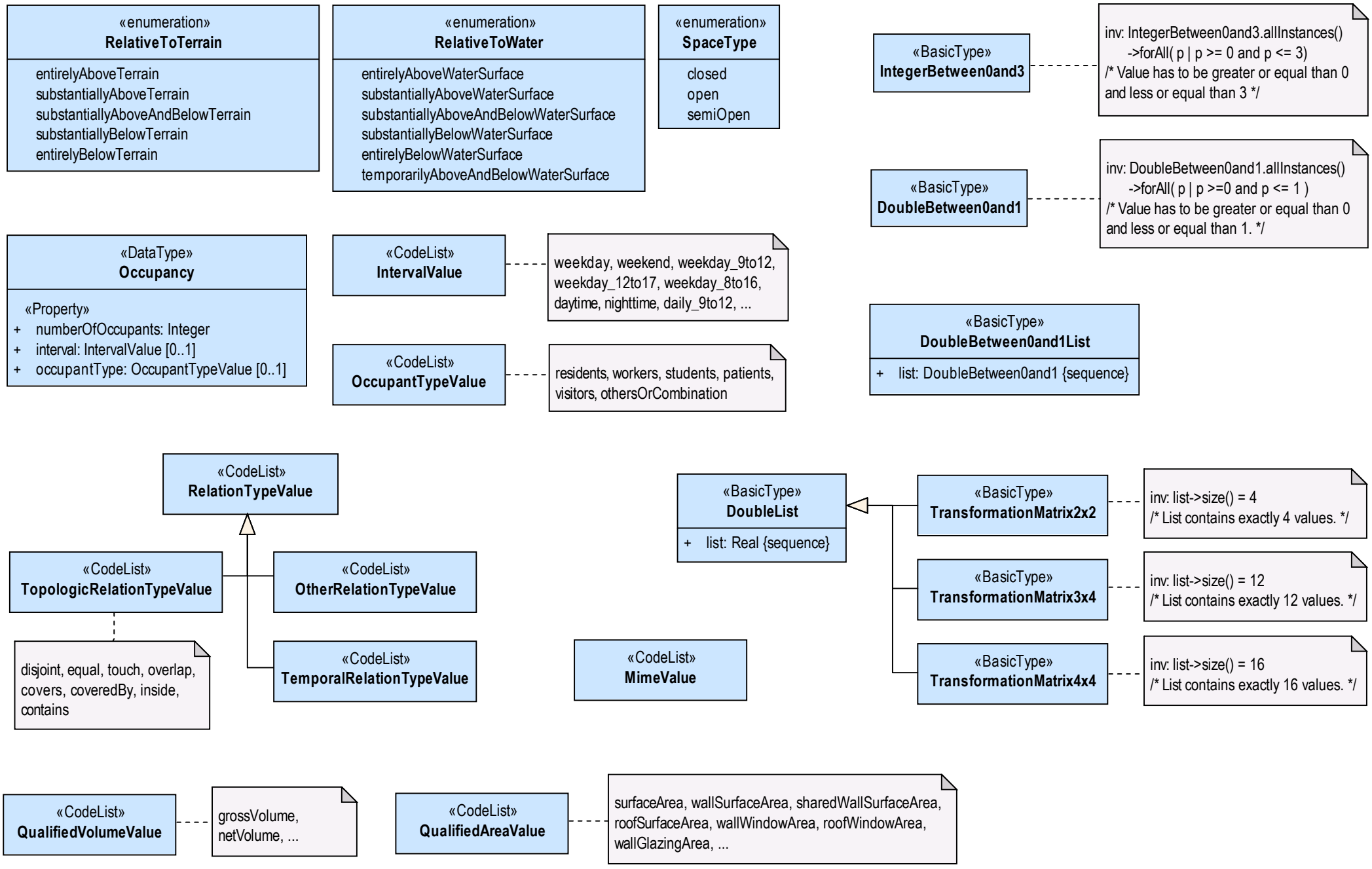
Core module - Miscellaneous



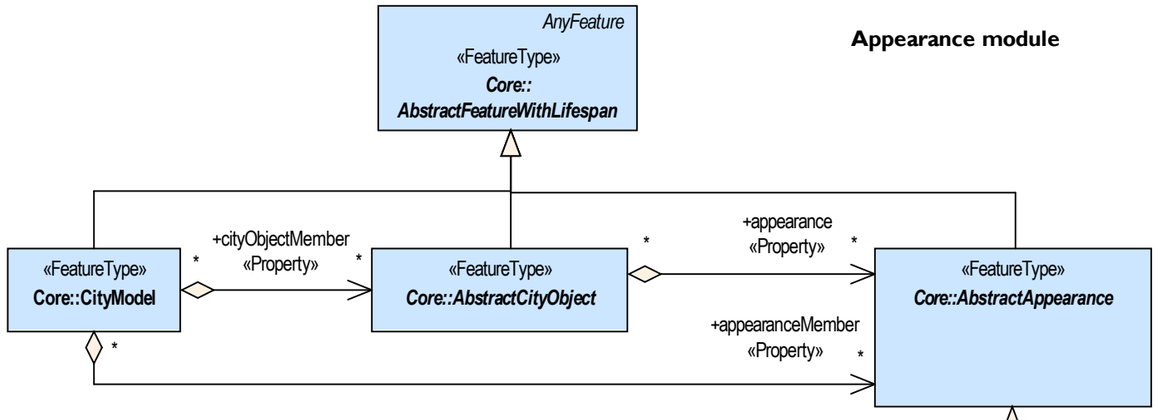
ExternalReference is now extended by an optional ***relationType*** which can link to some external definition of the type of relation (e.g. the *sameAs* relation from OWL). Hence, ***ExternalReferences*** can now be used to express relations similar to RDF.

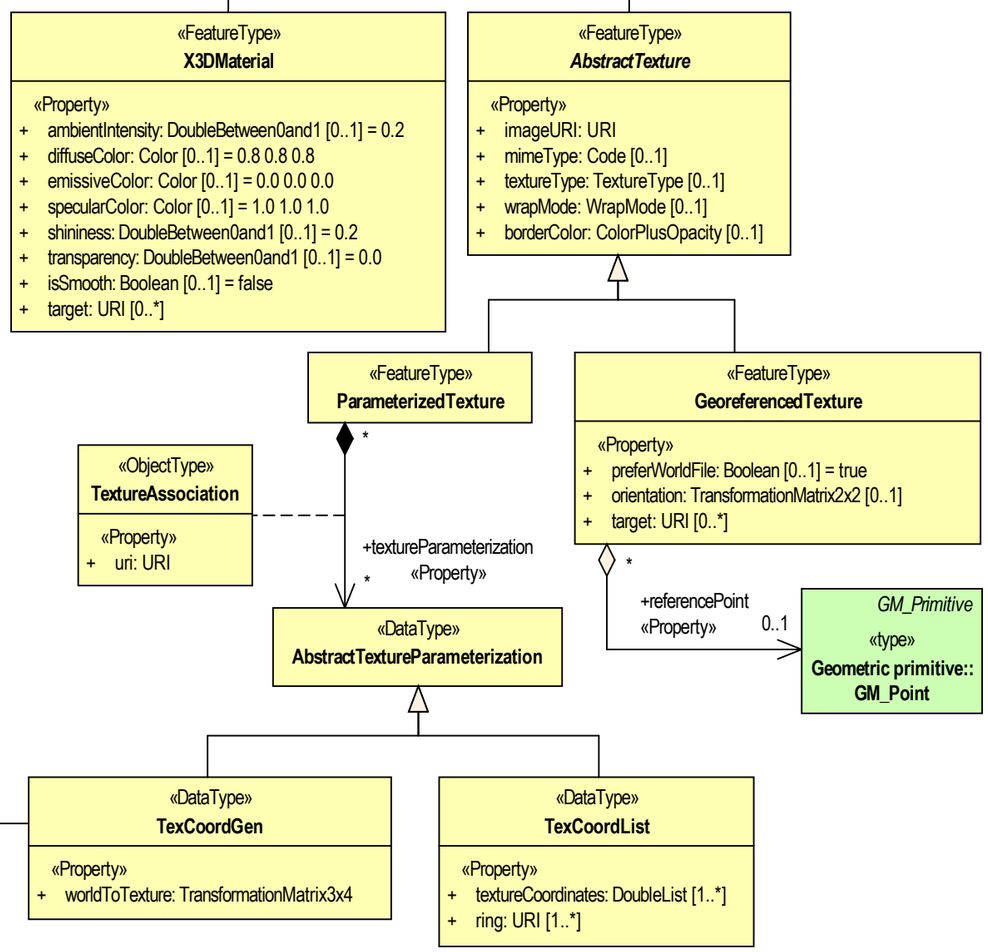


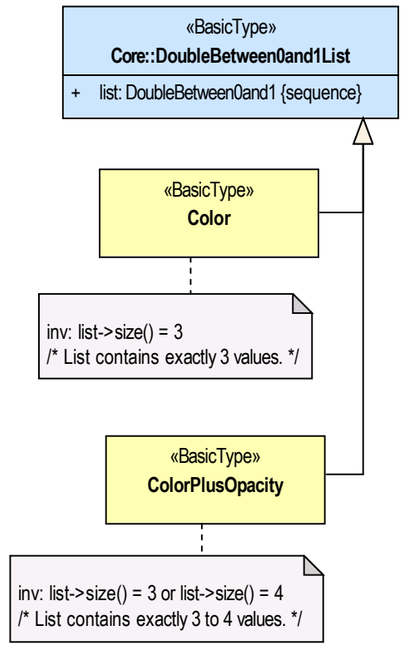
Core module - Basic Types, Enumerations, and Code lists

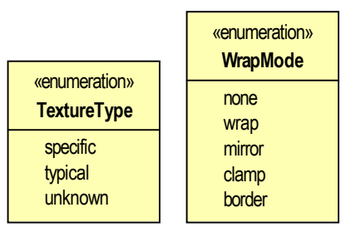


Appearance module

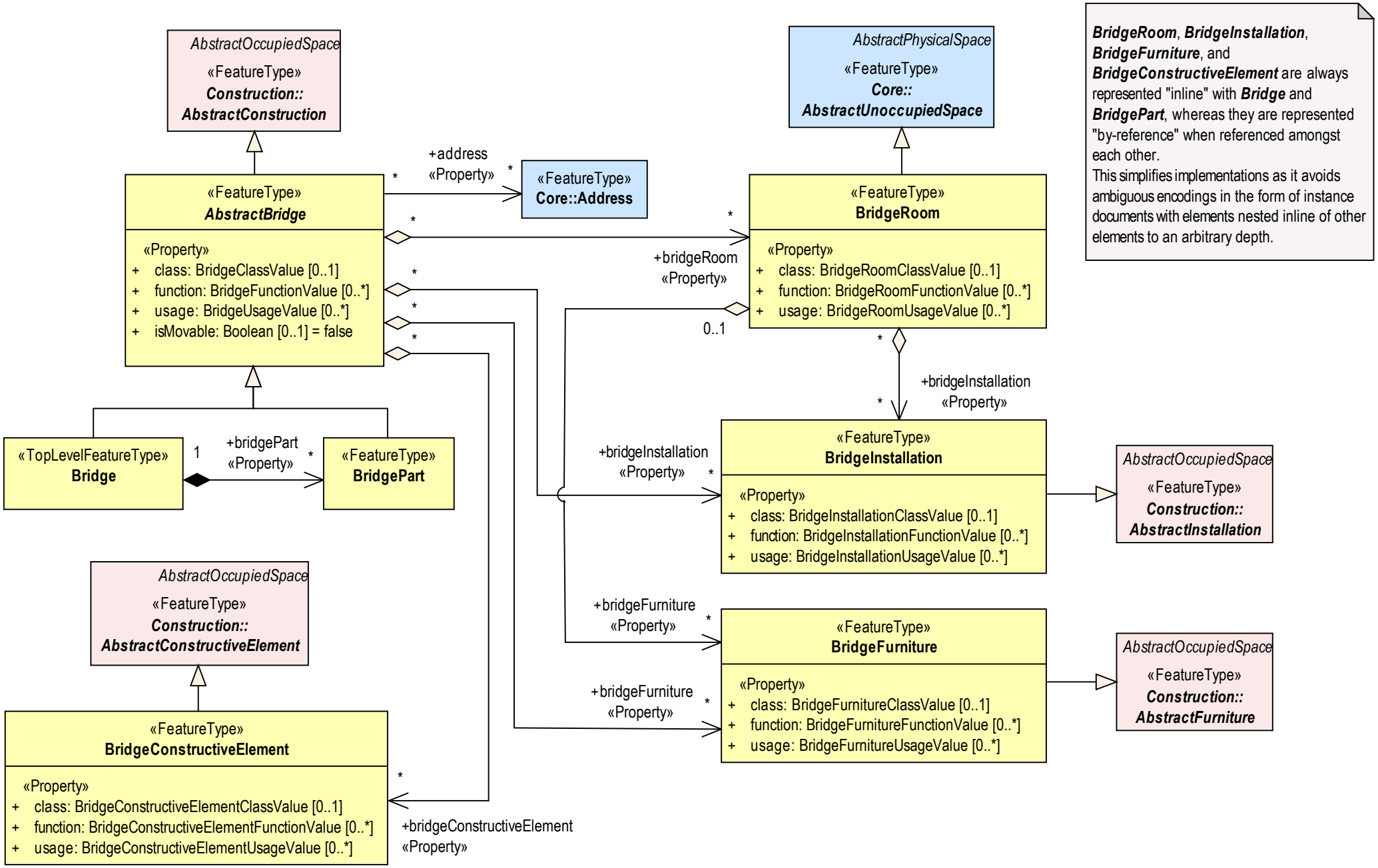




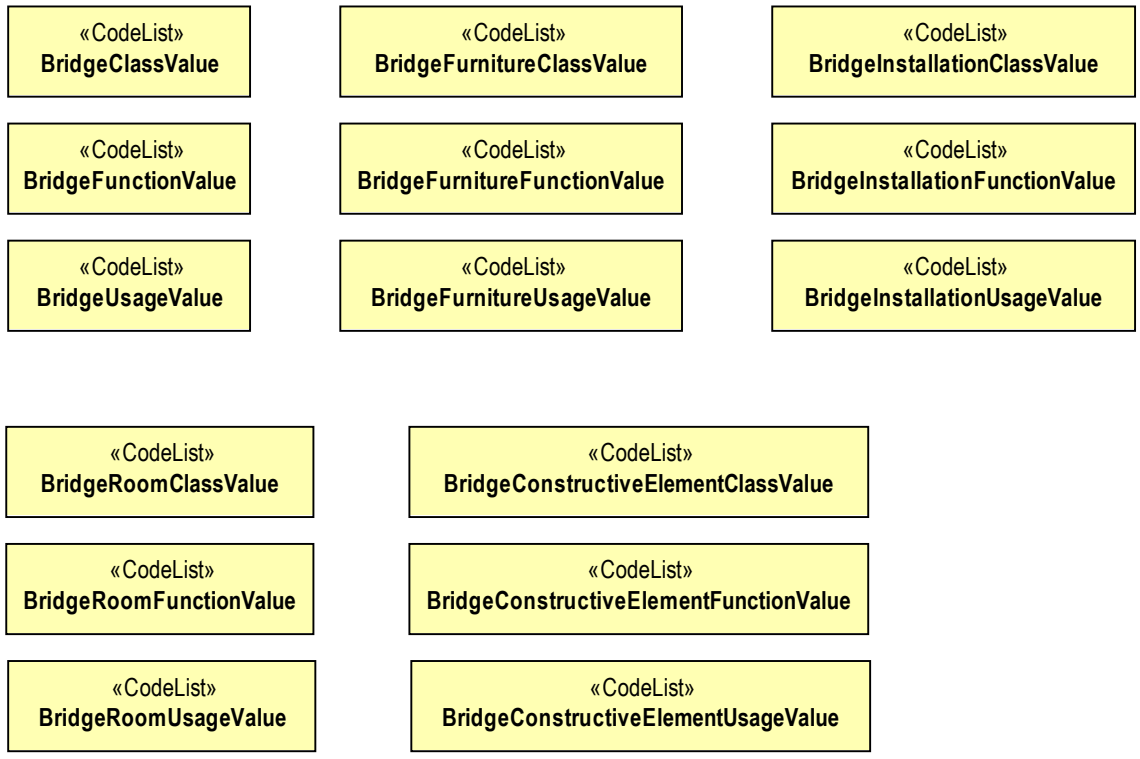




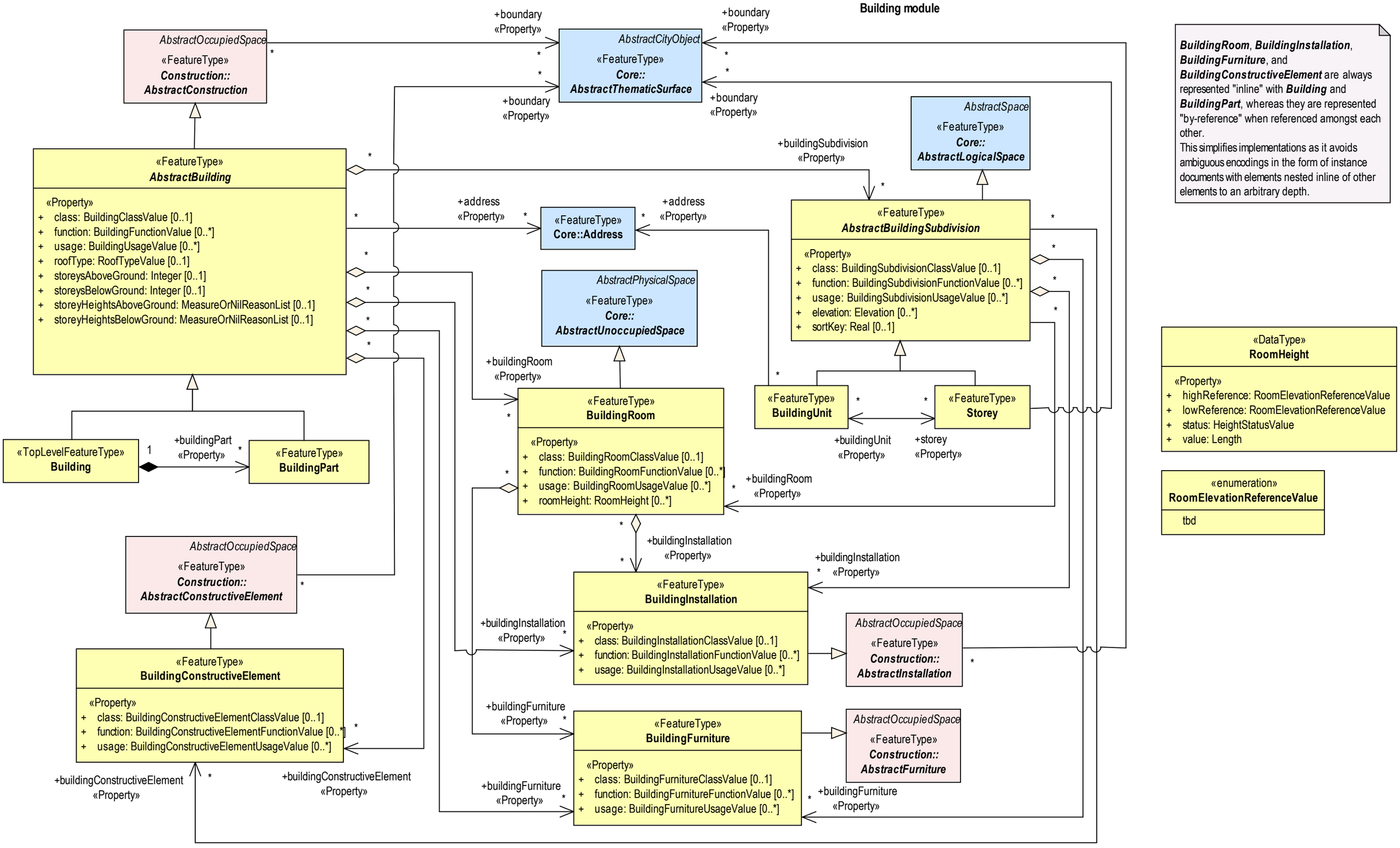
Bridge module



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.

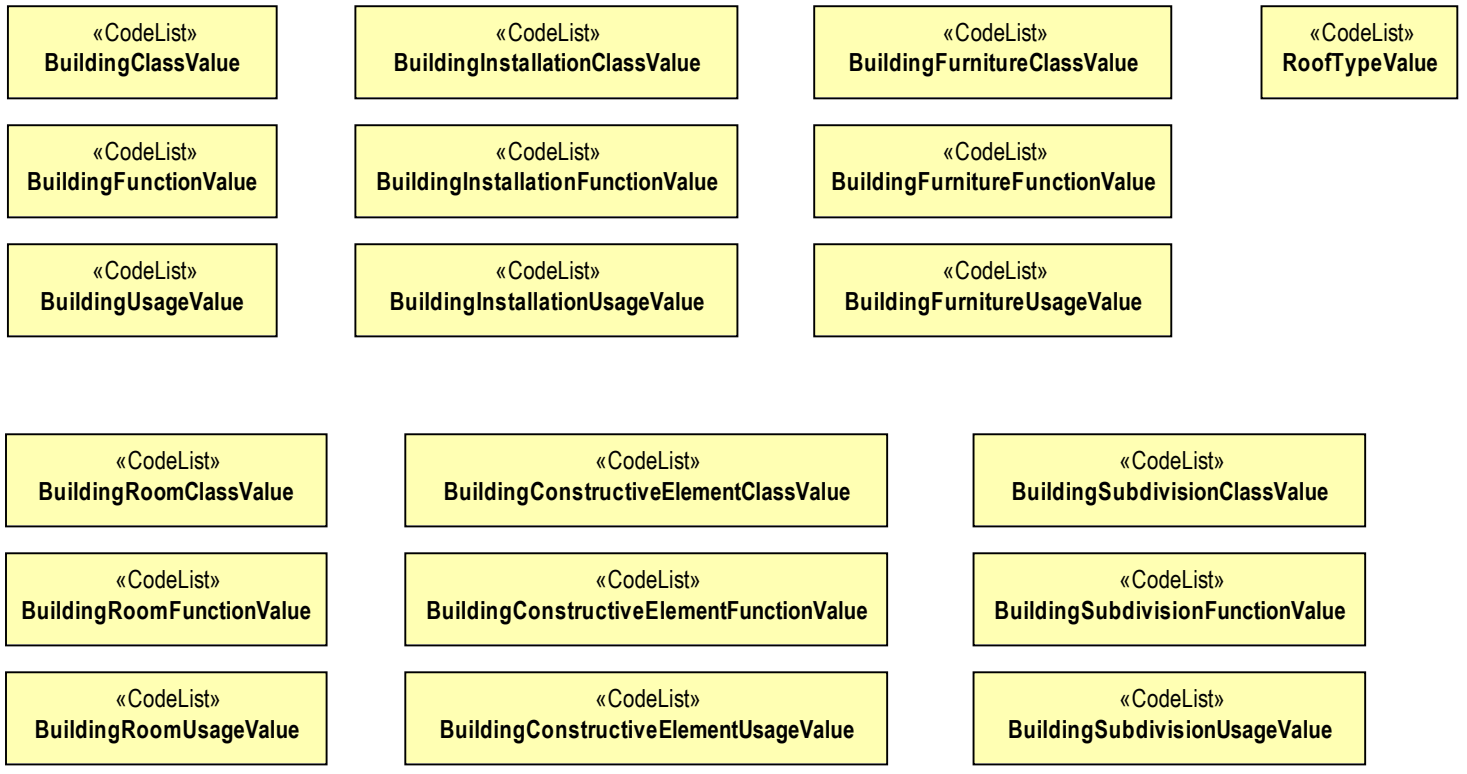
«DataType»
RoomHeight

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

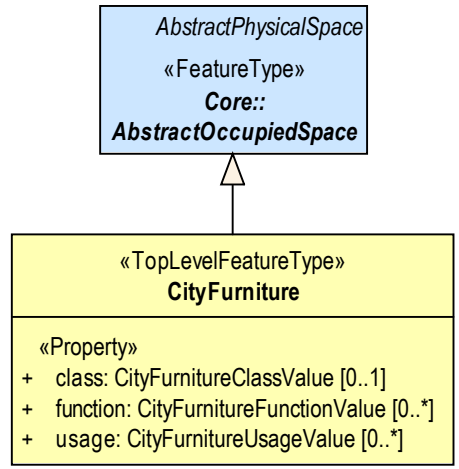
«enumeration»
RoomElevationReferenceValue

tbd

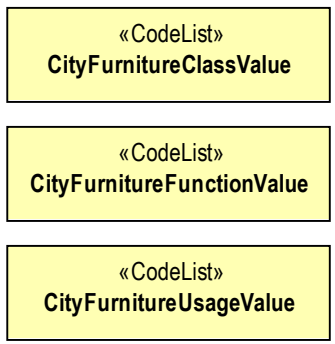
Building module - Code lists



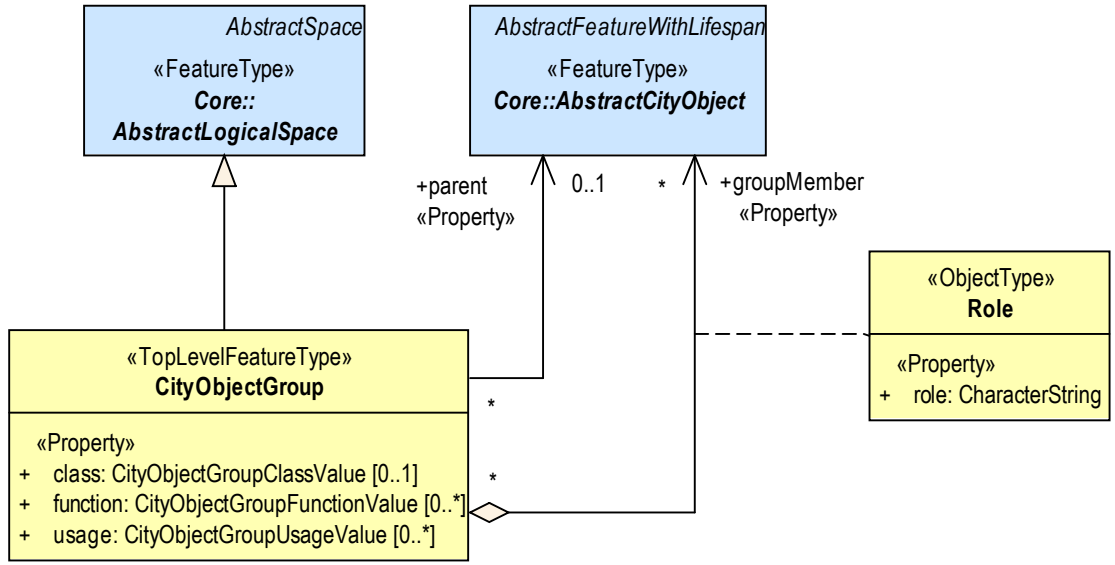
CityFurniture module



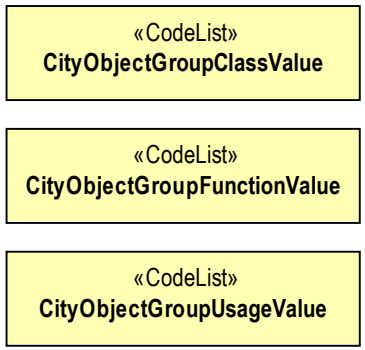
CityFurniture module - Code lists



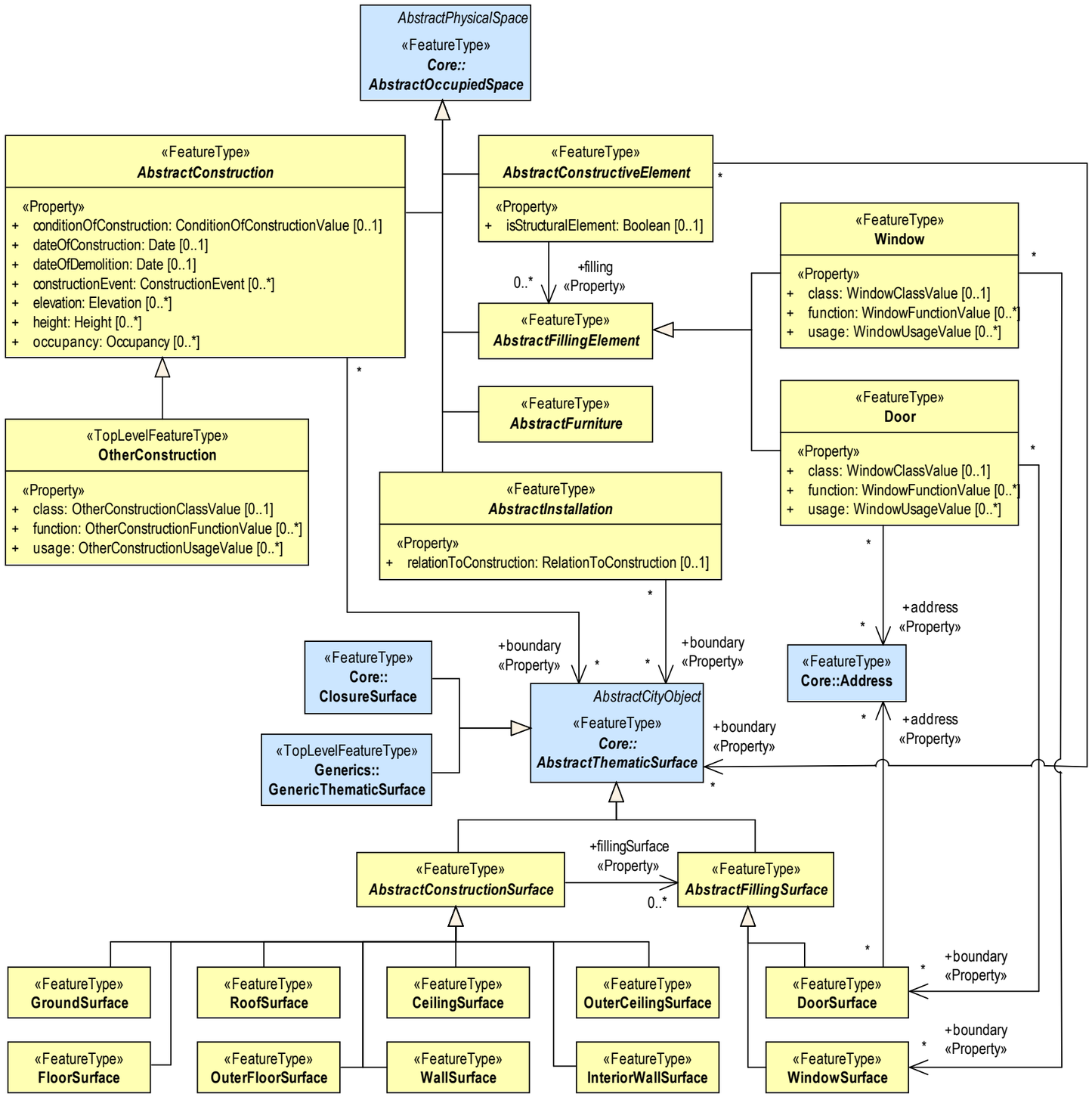
CityObjectGroup module



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

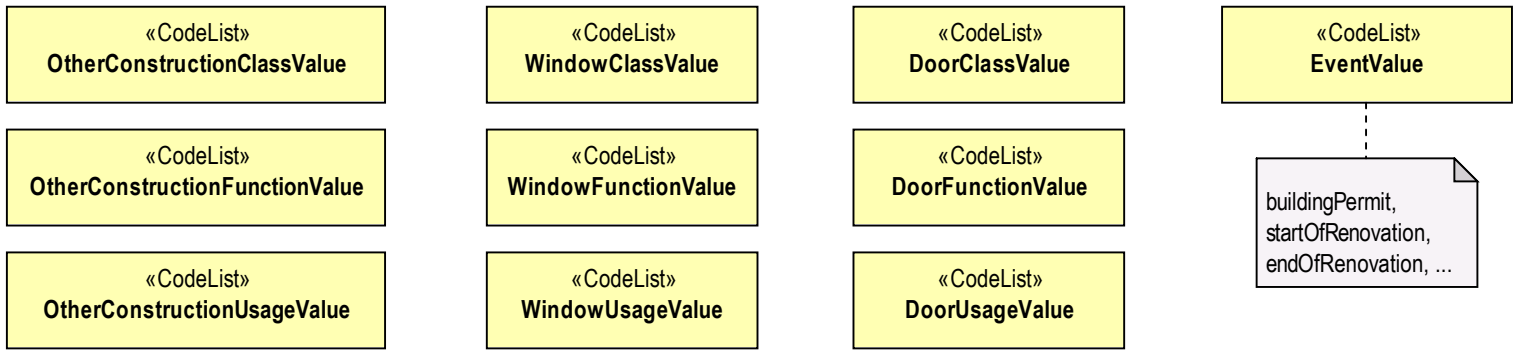
- estimated
- measured

Reference values for heights below ground will be added.

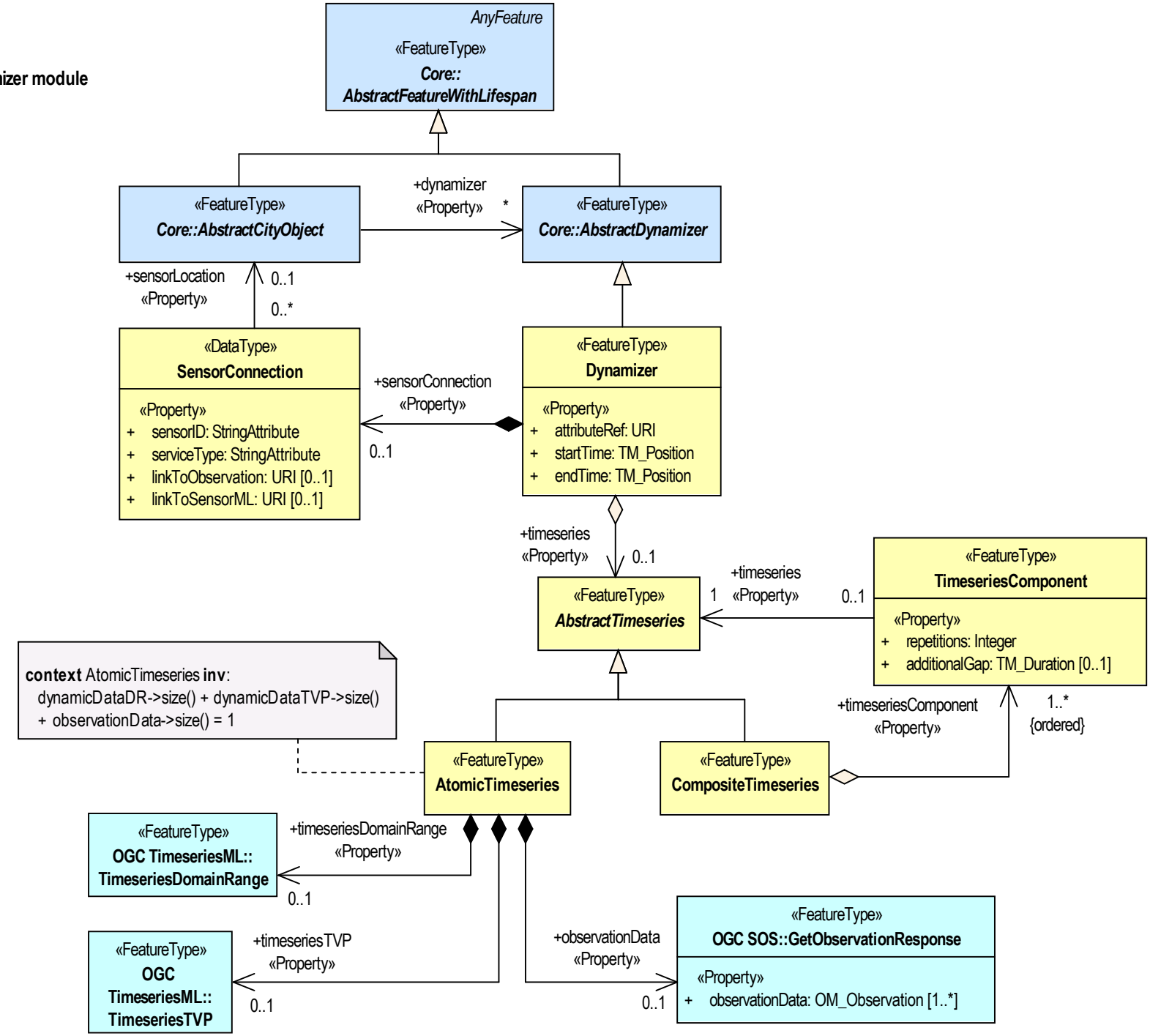
«DataType»
ConstructionEvent

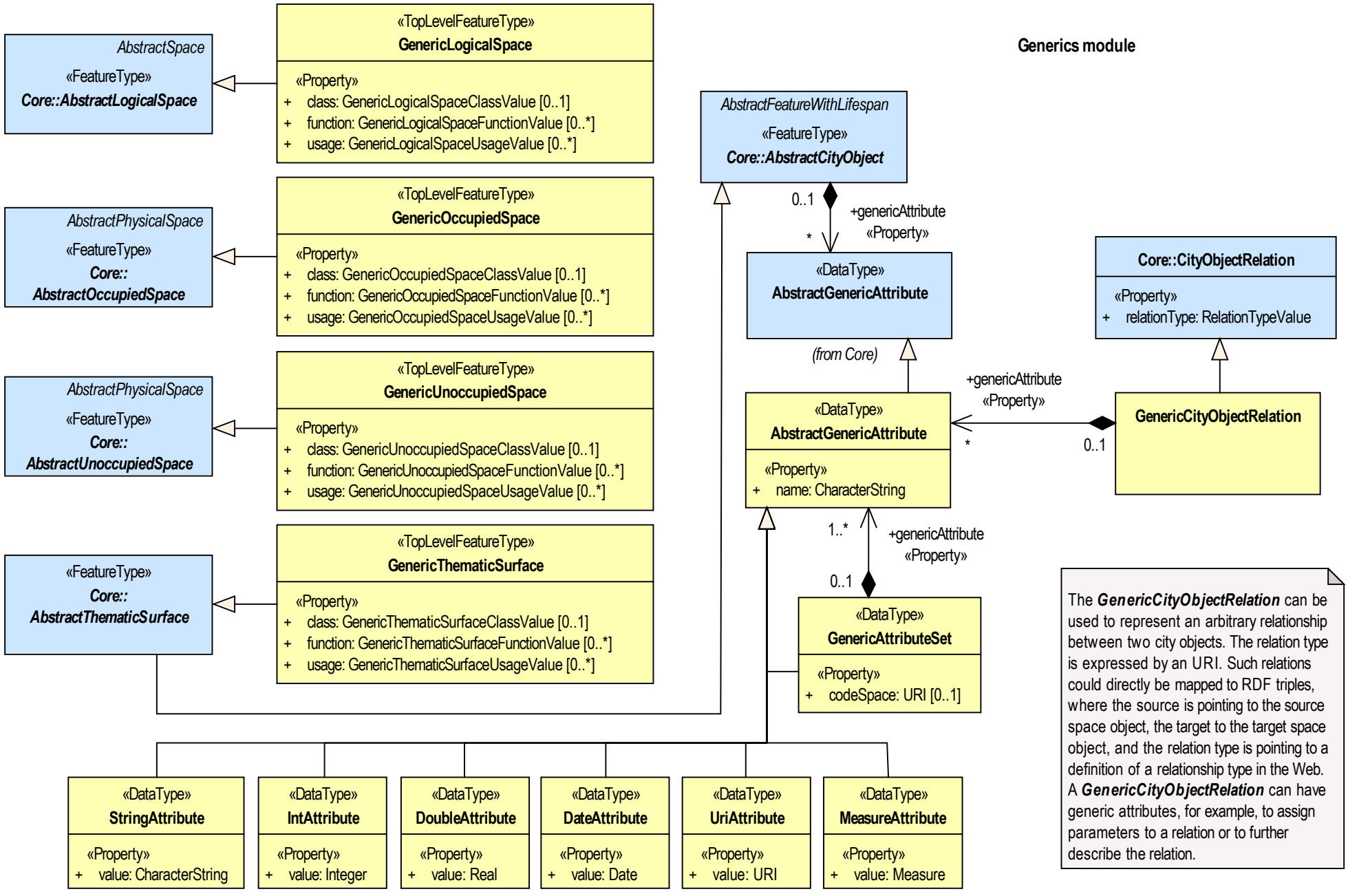
«Property»
+ event: EventValue
+ dateOfEvent: Date
+ description: CharacterString [0..1]

Construction module - Code lists

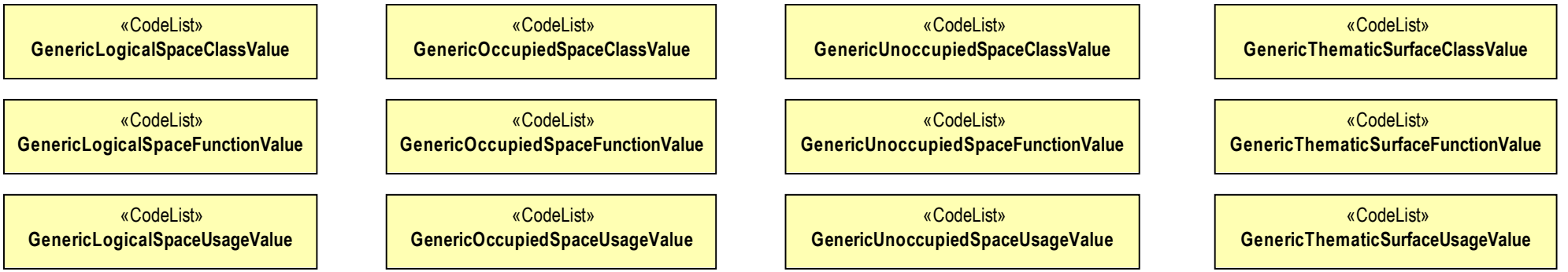


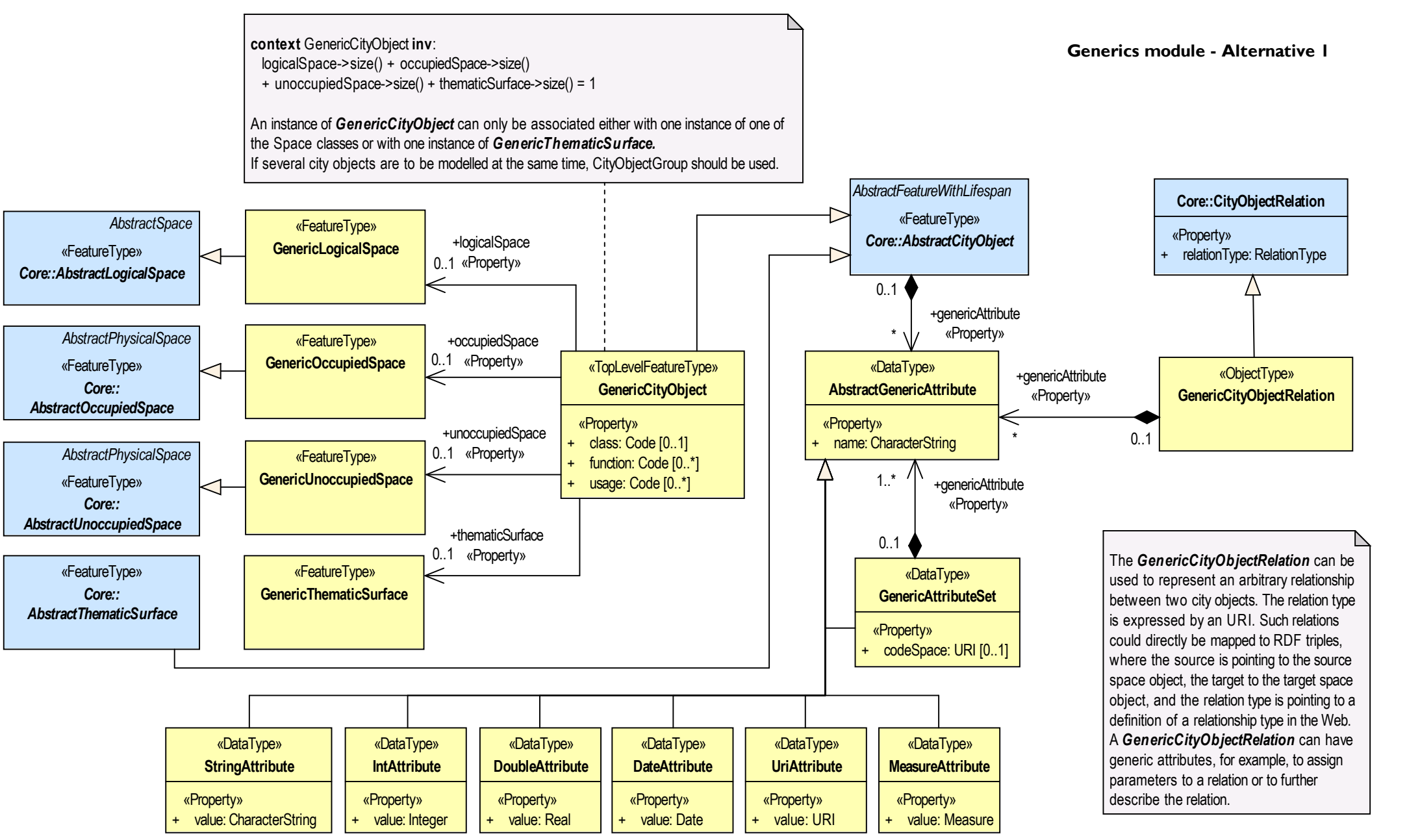
Dynamizer module



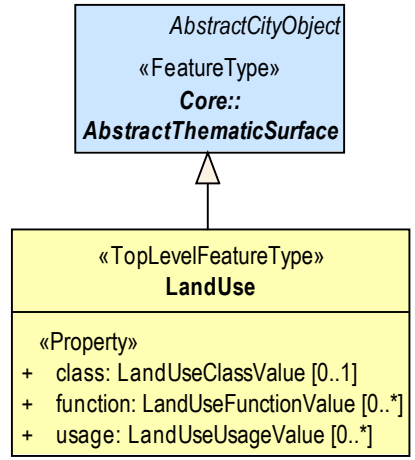

Generics module
AbstractSpace
«FeatureType»
Core::AbstractLogicalSpace
«TopLevelFeatureType»
GenericLogicalSpace
«Property»
+ class: GenericLogicalSpaceClassValue [0..1]
+ function: GenericLogicalSpaceFunctionValue [0..*]
+ usage: GenericLogicalSpaceUsageValue [0..*]
AbstractPhysicalSpace
«FeatureType»
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
(from Core)
«DataType»
AbstractGenericAttribute
«Property»
+ name: CharacterString
+genericAttribute
«Property»
*
0..1
Core::CityObjectRelation
«Property»
+ relationType: RelationTypeValue
GenericCityObjectRelation
1..*
0..1
+genericAttribute
«Property»
«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
The GenericCityObjectRelation can be used to represent an arbitrary relationship between two city objects. The relation type is expressed by an URI. Such relations could directly be mapped to RDF triples, where the source is pointing to the source space object, the target to the target space object, and the relation type is pointing to a definition of a relationship type in the Web. A GenericCityObjectRelation can have generic attributes, for example, to assign parameters to a relation or to further describe the relation.

Generics module - Code lists

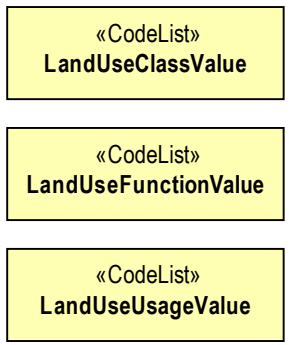



context GenericCityObject inv:
logicalSpace->size() + occupiedSpace->size()
+ unoccupiedSpace->size() + thematicSurface->size() = 1
An instance of GenericCityObject can only be associated either with one instance of one of the Space classes or with one instance of GenericThematicSurface.
If several city objects are to be modelled at the same time, CityObjectGroup should be used.
Generics module - Alternative I
AbstractSpace
«FeatureType»
Core::AbstractLogicalSpace
«FeatureType»
GenericLogicalSpace
+logicalSpace
0..1 «Property»
AbstractPhysicalSpace
«FeatureType»
Core::
AbstractOccupiedSpace
«FeatureType»
GenericOccupiedSpace
+occupiedSpace
0..1 «Property»
AbstractPhysicalSpace
«FeatureType»
Core::
AbstractUnoccupiedSpace
«FeatureType»
GenericUnoccupiedSpace
+unoccupiedSpace
0..1 «Property»
«FeatureType»
Core::
AbstractThematicSurface
«FeatureType»
GenericThematicSurface
+thematicSurface
0..1 «Property»
«TopLevelFeatureType»
GenericCityObject
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
AbstractFeatureWithLifespan
«FeatureType»
Core::AbstractCityObject
0..1
*
+genericAttribute
«Property»
«DataType»
AbstractGenericAttribute
«Property»
+ name: CharacterString
+genericAttribute
«Property»
*
1..*
+genericAttribute
«Property»
0..1
«DataType»
GenericAttributeSet
«Property»
+ codeSpace: URI [0..1]
Core::CityObjectRelation
«Property»
+ relationType: RelationType
«ObjectType»
GenericCityObjectRelation
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
The GenericCityObjectRelation can be used to represent an arbitrary relationship between two city objects. The relation type is expressed by an URI. Such relations could directly be mapped to RDF triples, where the source is pointing to the source space object, the target to the target space object, and the relation type is pointing to a definition of a relationship type in the Web. A GenericCityObjectRelation can have generic attributes, for example, to assign parameters to a relation or to further describe the relation.

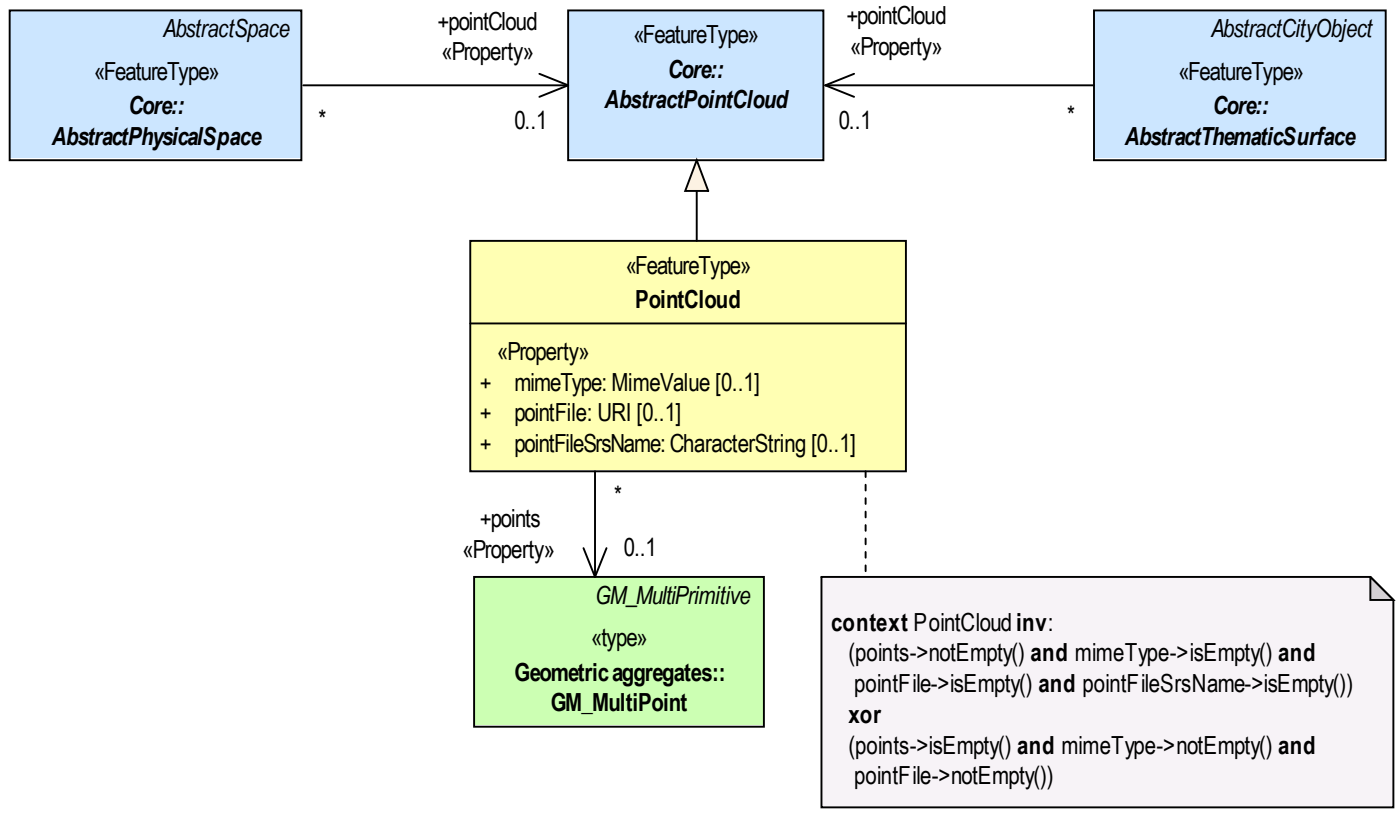
LandUse module

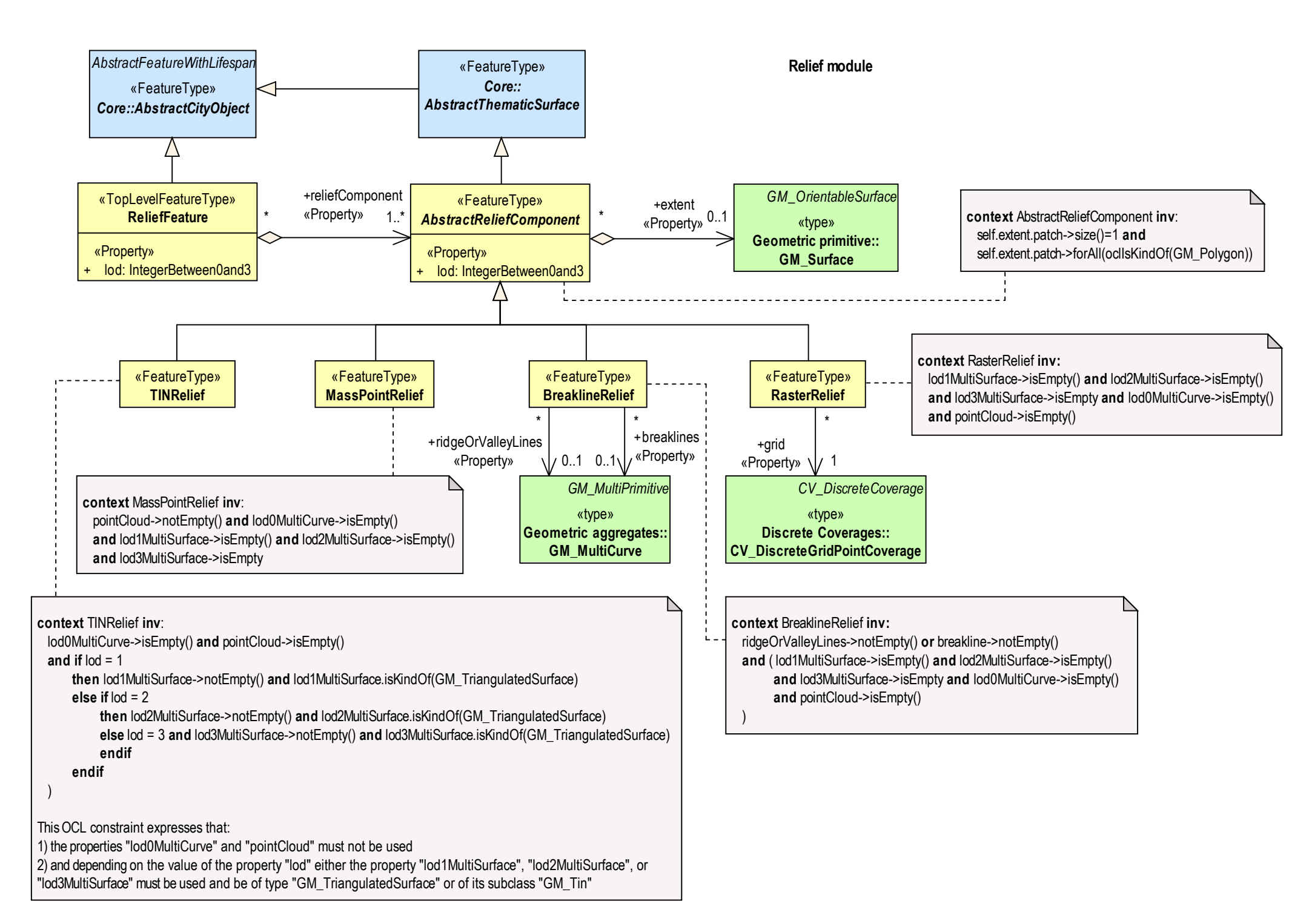


LandUse module - Code lists



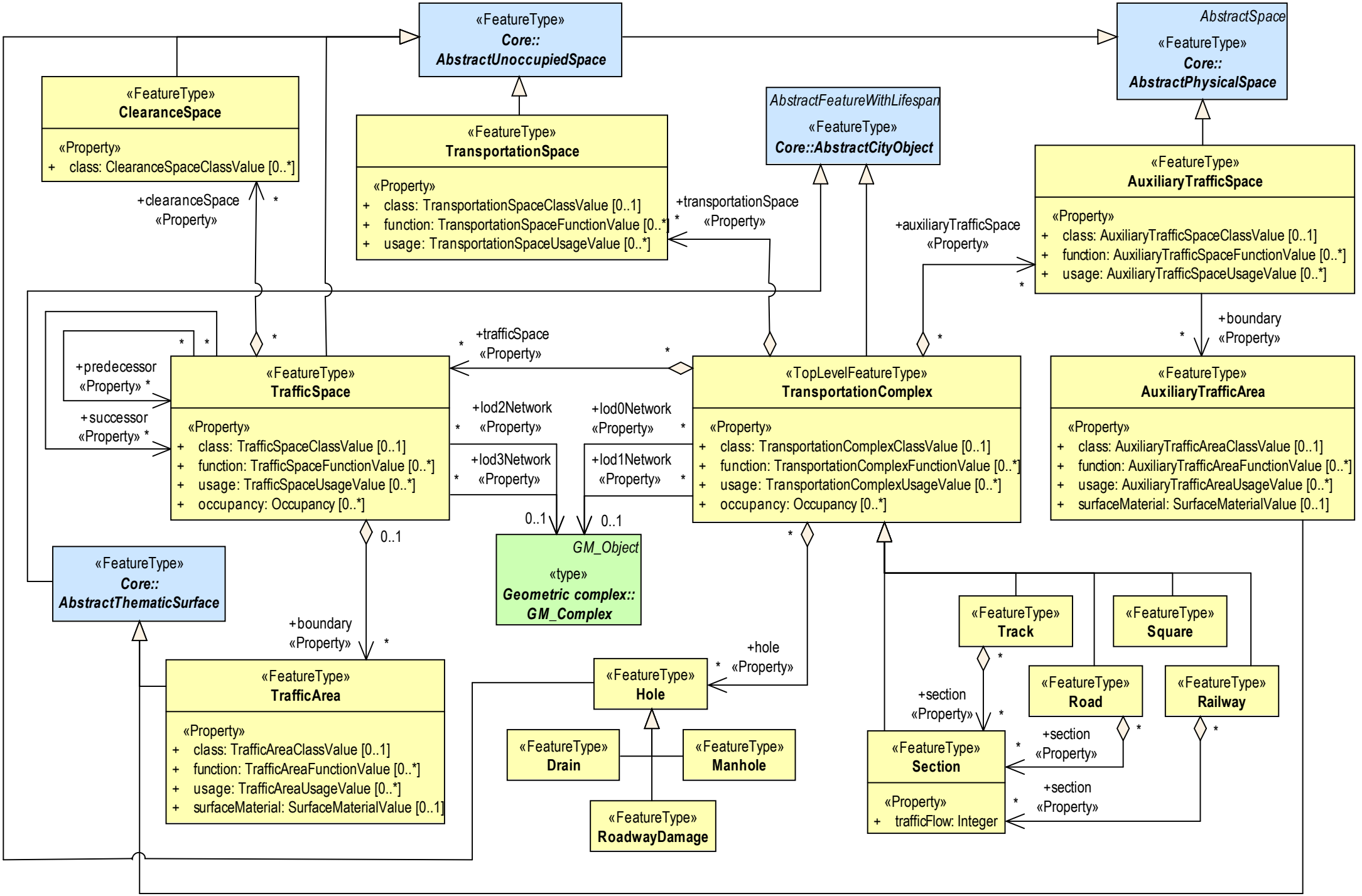
PointCloud module



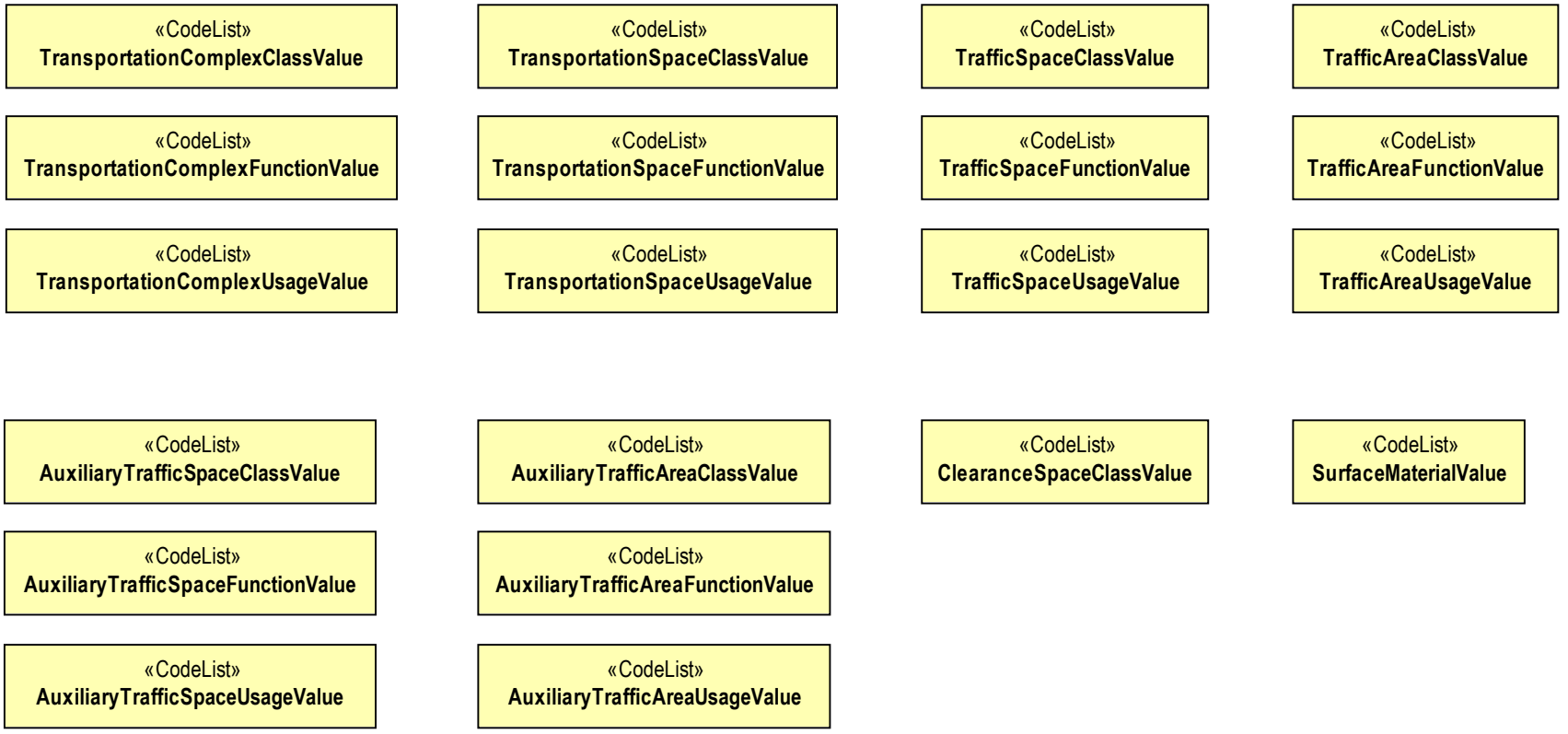

Relief module
AbstractFeatureWithLifespan
«FeatureType»
Core::AbstractCityObject
«FeatureType»
Core::
AbstractThematicSurface
«TopLevelFeatureType»
ReliefFeature
«Property»
+ lod: IntegerBetween0and3
+reliefComponent
«Property»
1..*
«FeatureType»
AbstractReliefComponent
«Property»
+ lod: IntegerBetween0and3
+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()
)

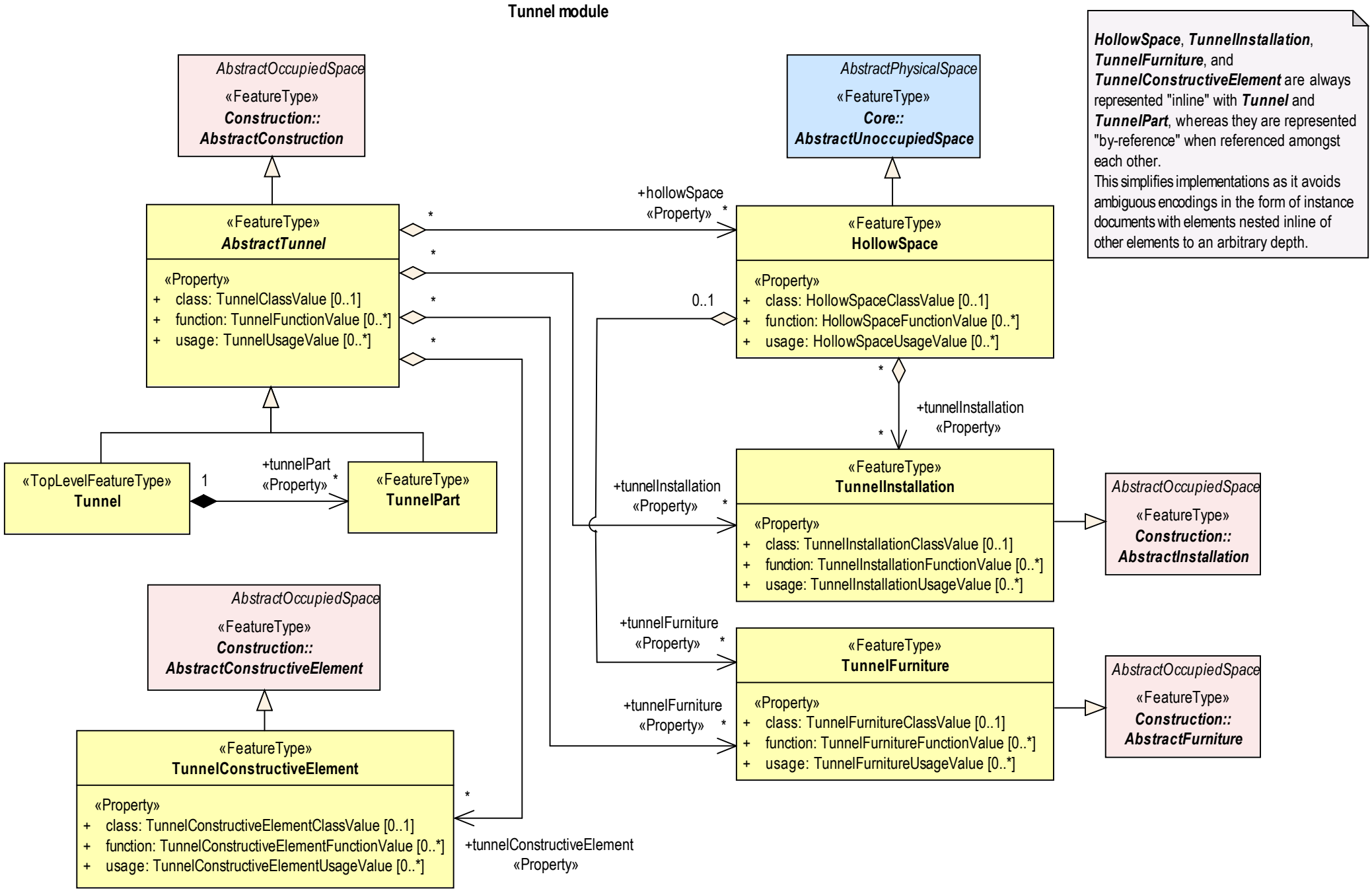
Transportation module

At the 3DGeoInfo conference at TU Delft early October 2018, a discussion between Anna Labetski and Thomas H. Kolbe took place regarding some refinements proposed by TU Delft in a conference paper. These refinements are going to be integrated. The current diagram only reflects the refined modelling of the TopLevelFeatureType concept.

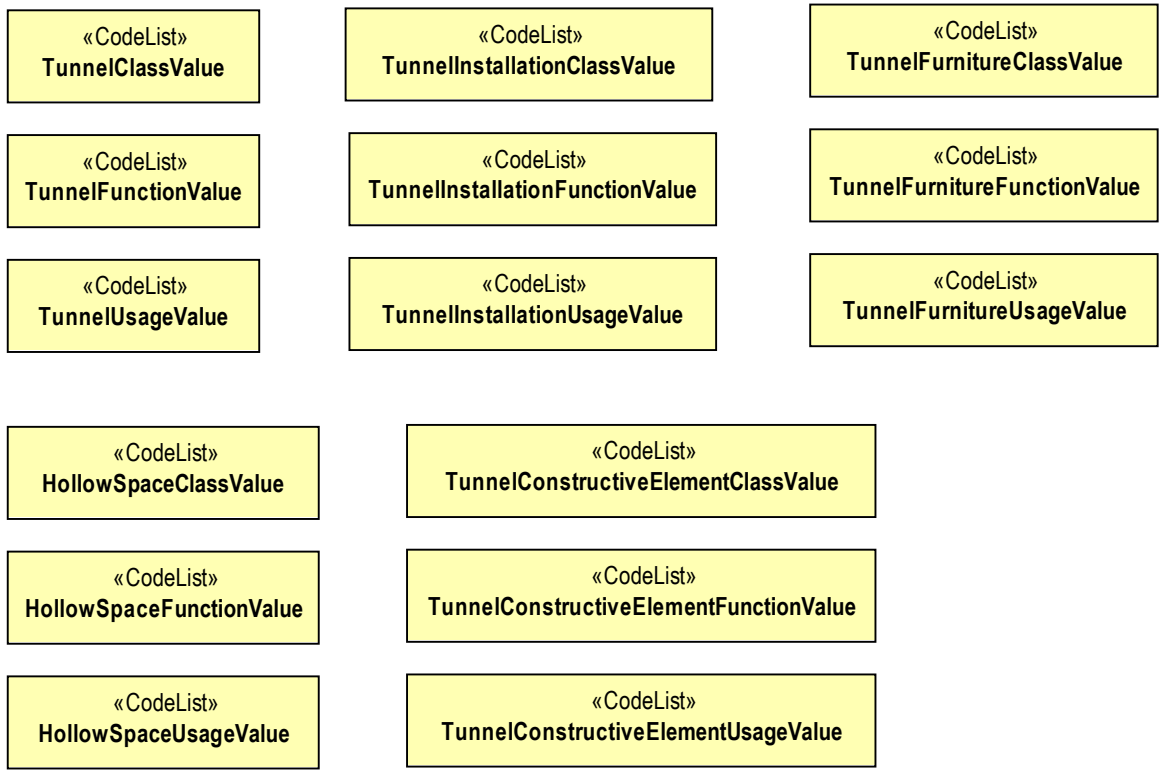


Transportation module - Code lists

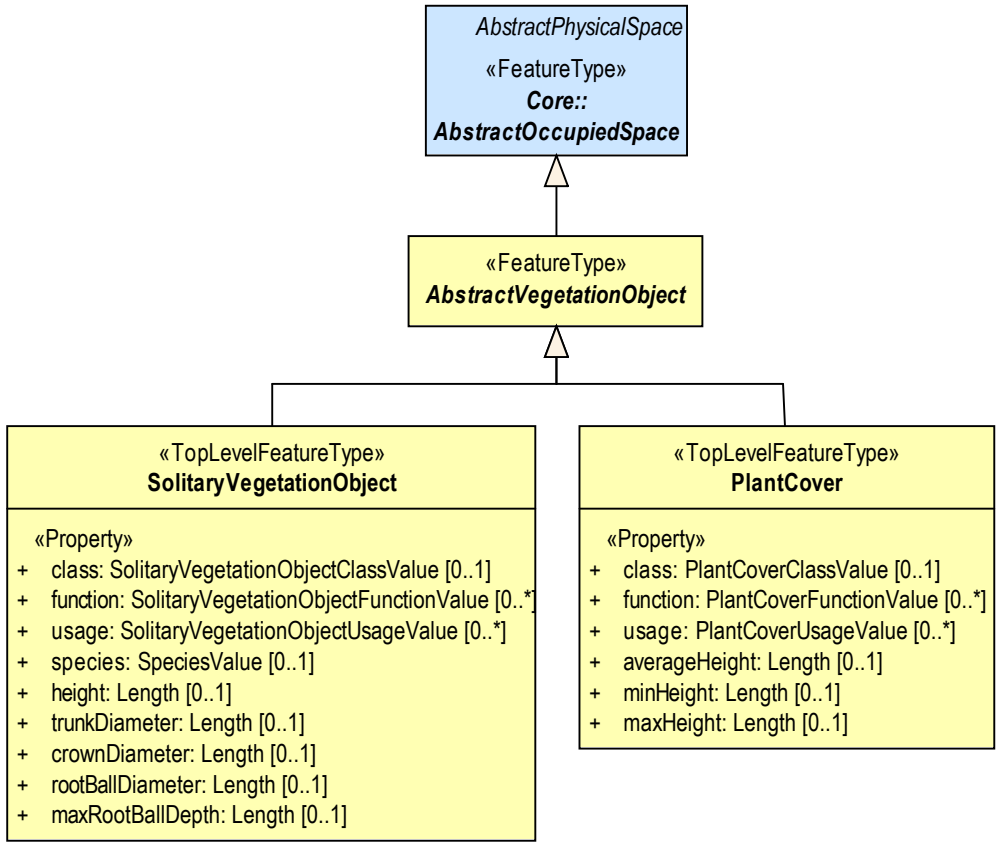



Tunnel module
AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractConstruction
AbstractPhysicalSpace
«FeatureType»
Core::
AbstractUnoccupiedSpace
HollowSpace, TunnelInstallation, TunnelFurniture, and TunnelConstructiveElement are always represented "inline" with Tunnel and TunnelPart, whereas 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»

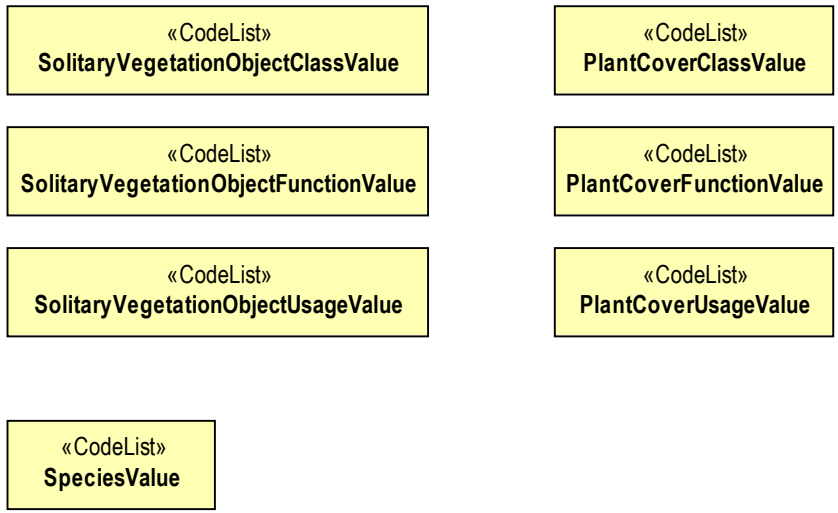
Tunnel module - Code lists



Vegetation module

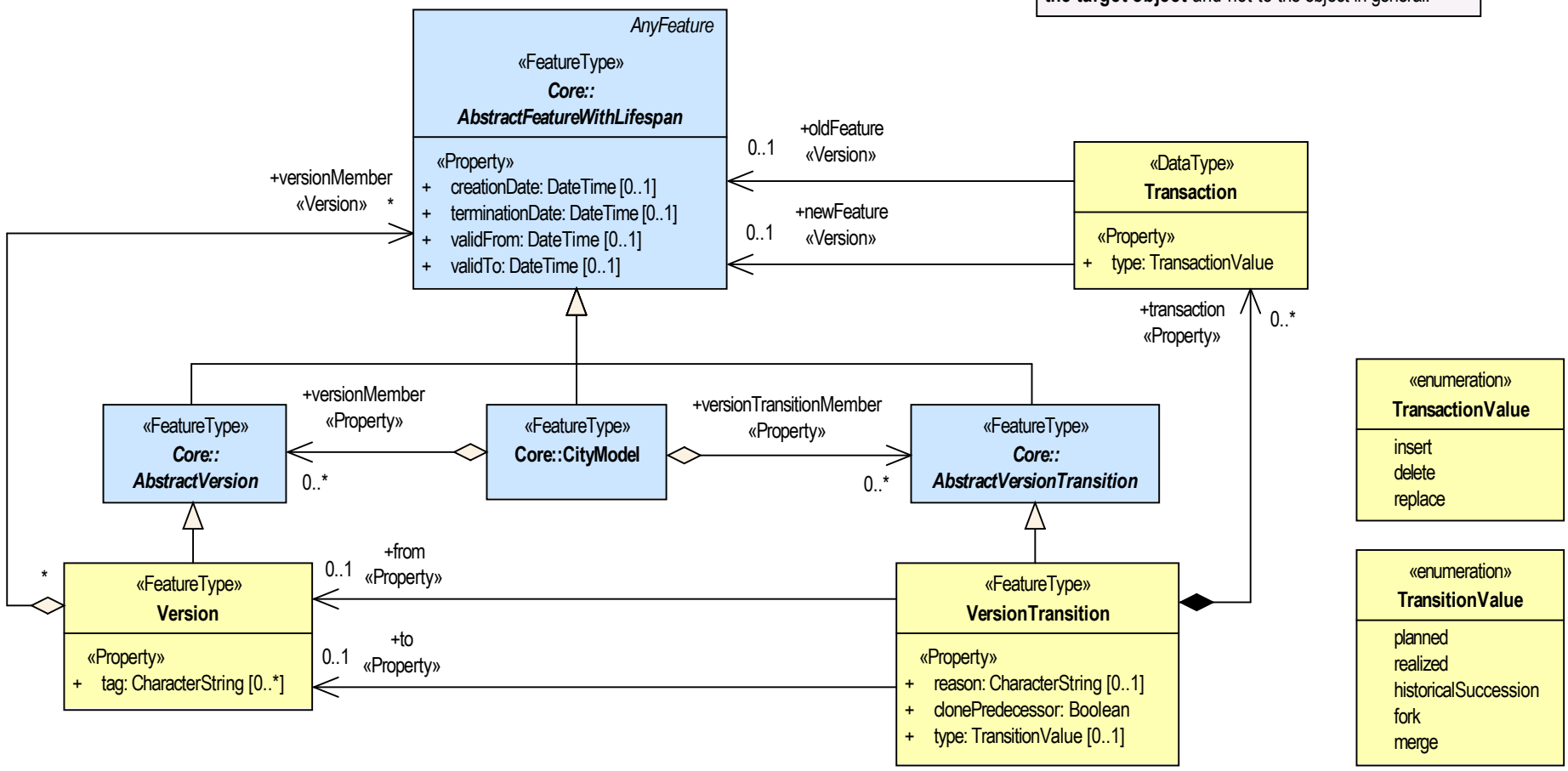


Vegetation module - Code lists

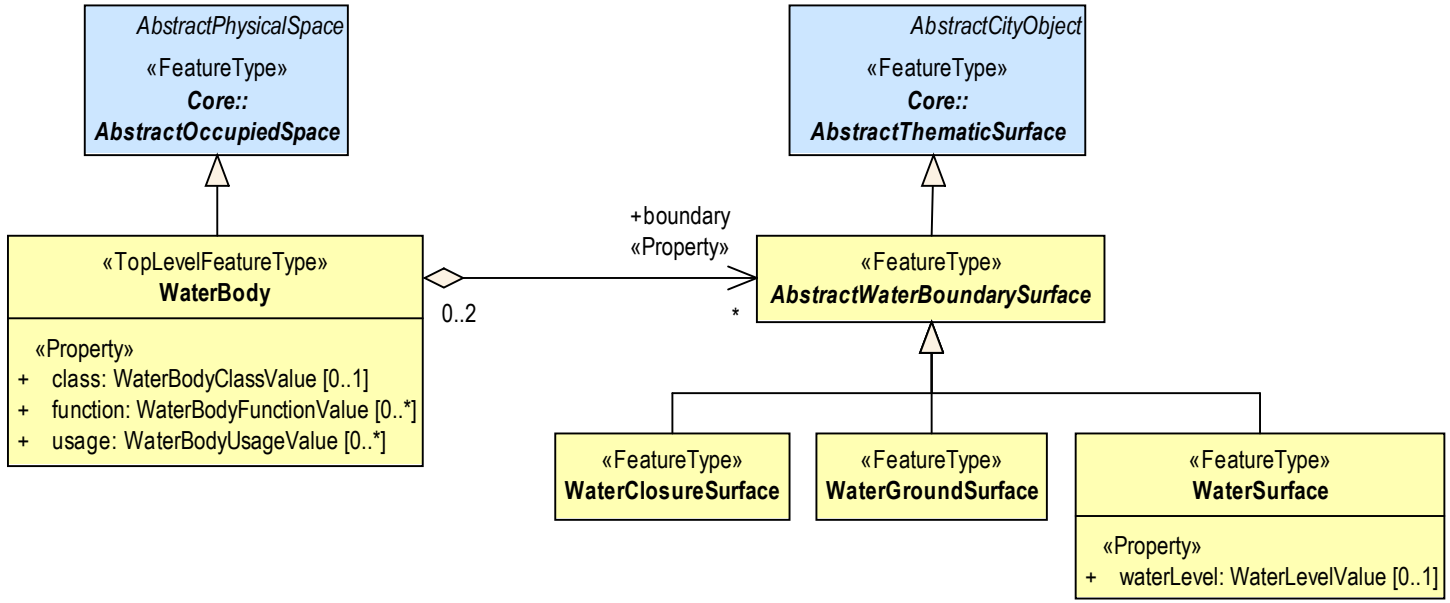


Versioning module

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