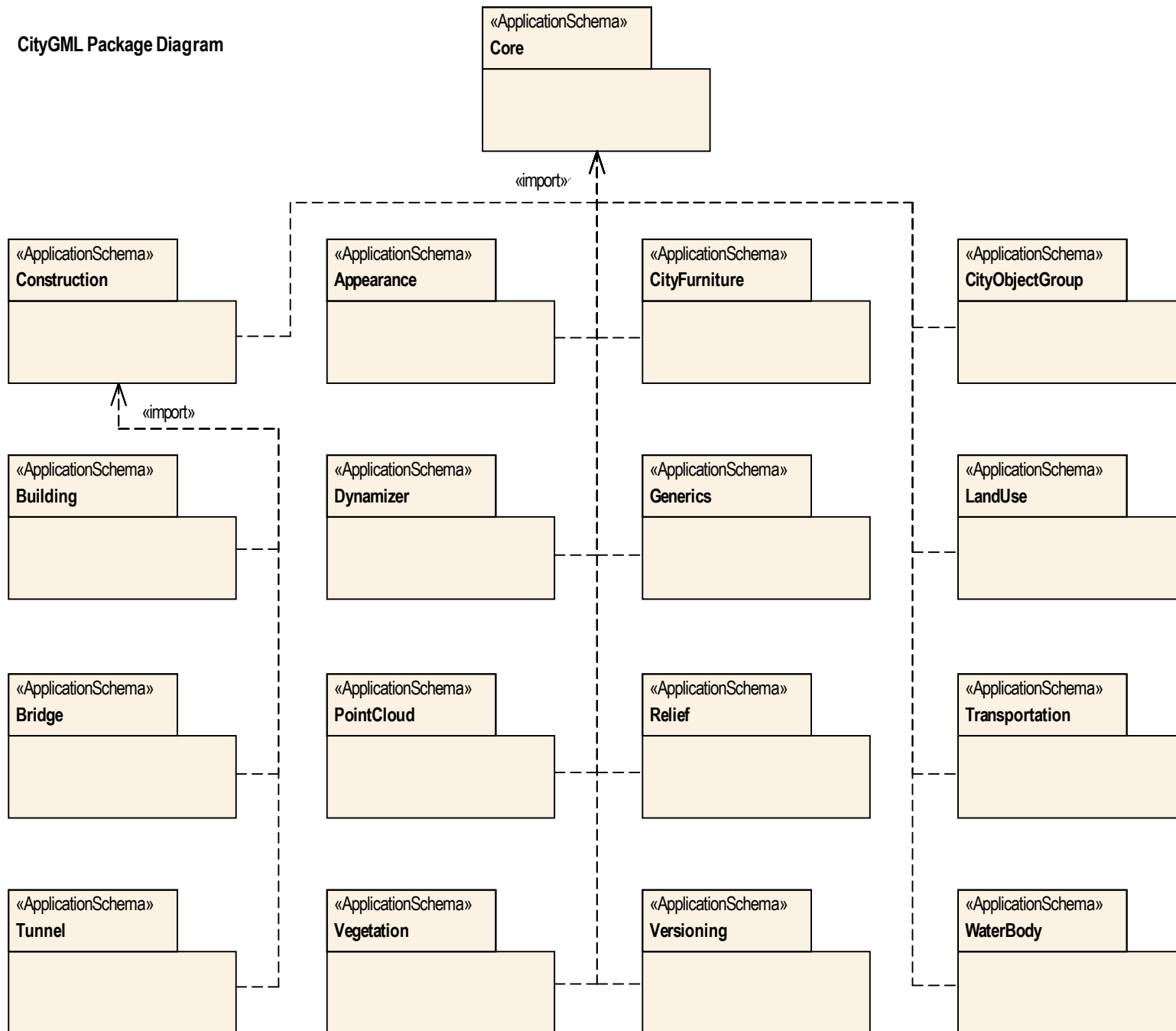
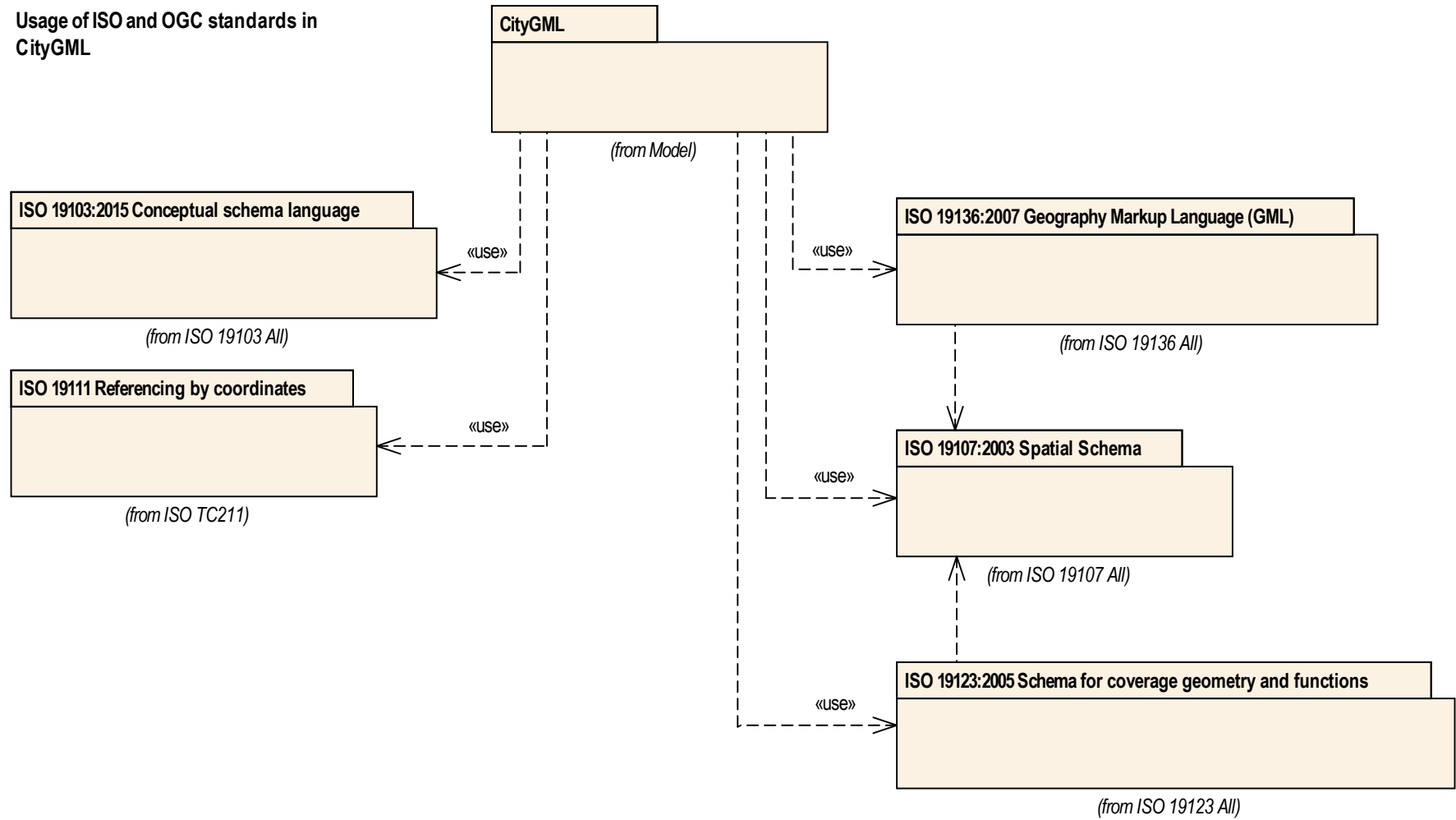
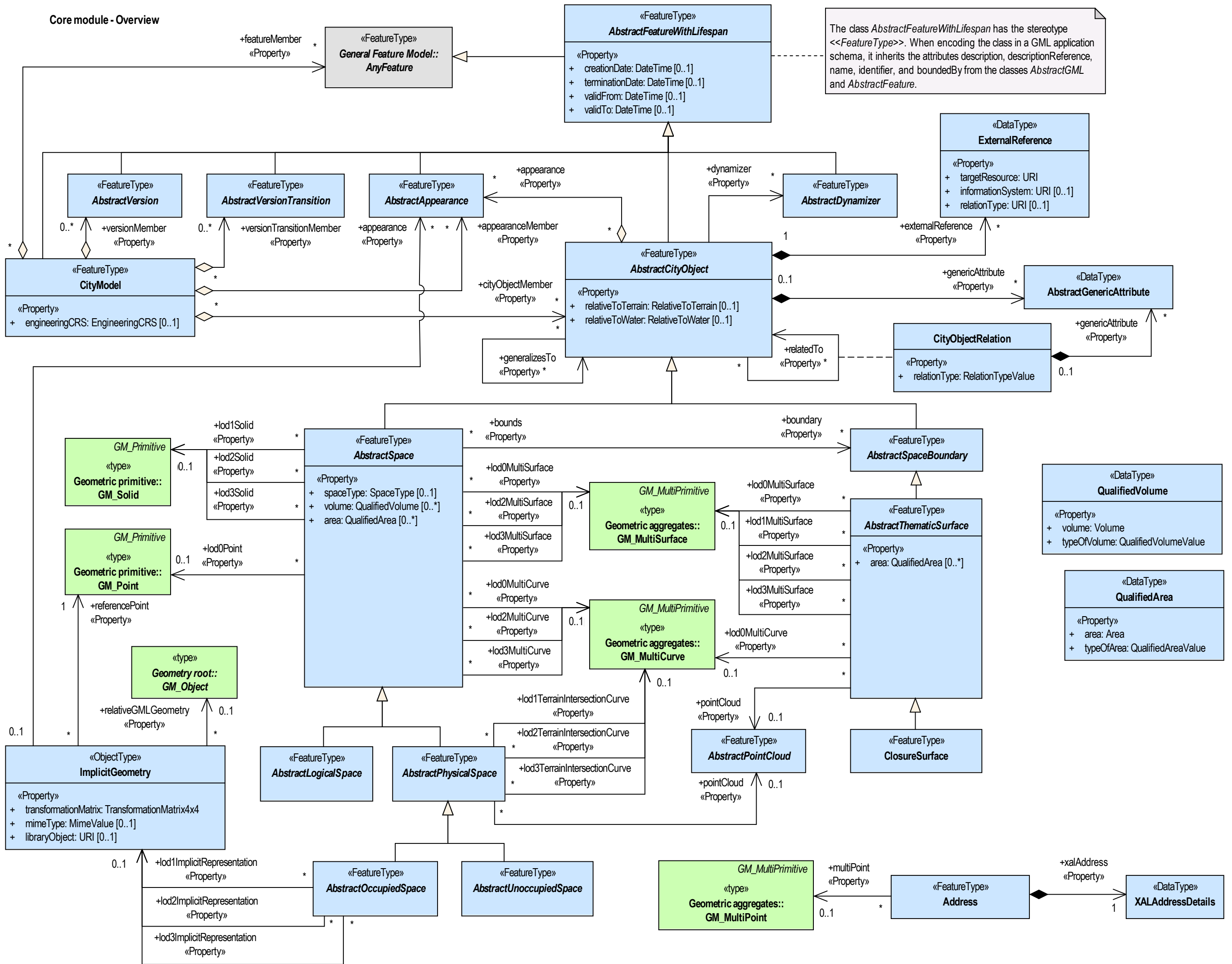


CityGML Package Diagram

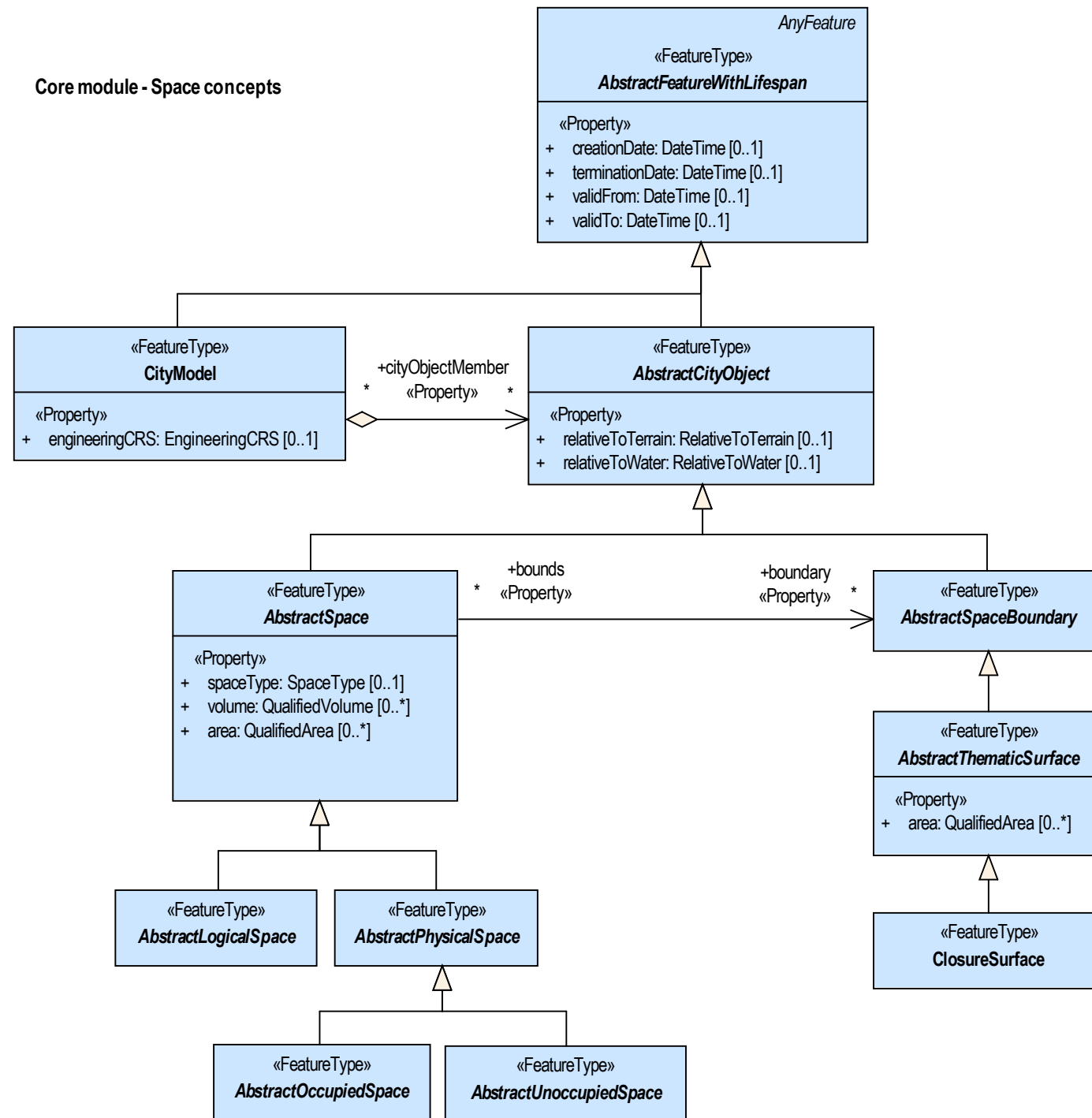


Usage of ISO and OGC standards in CityGML

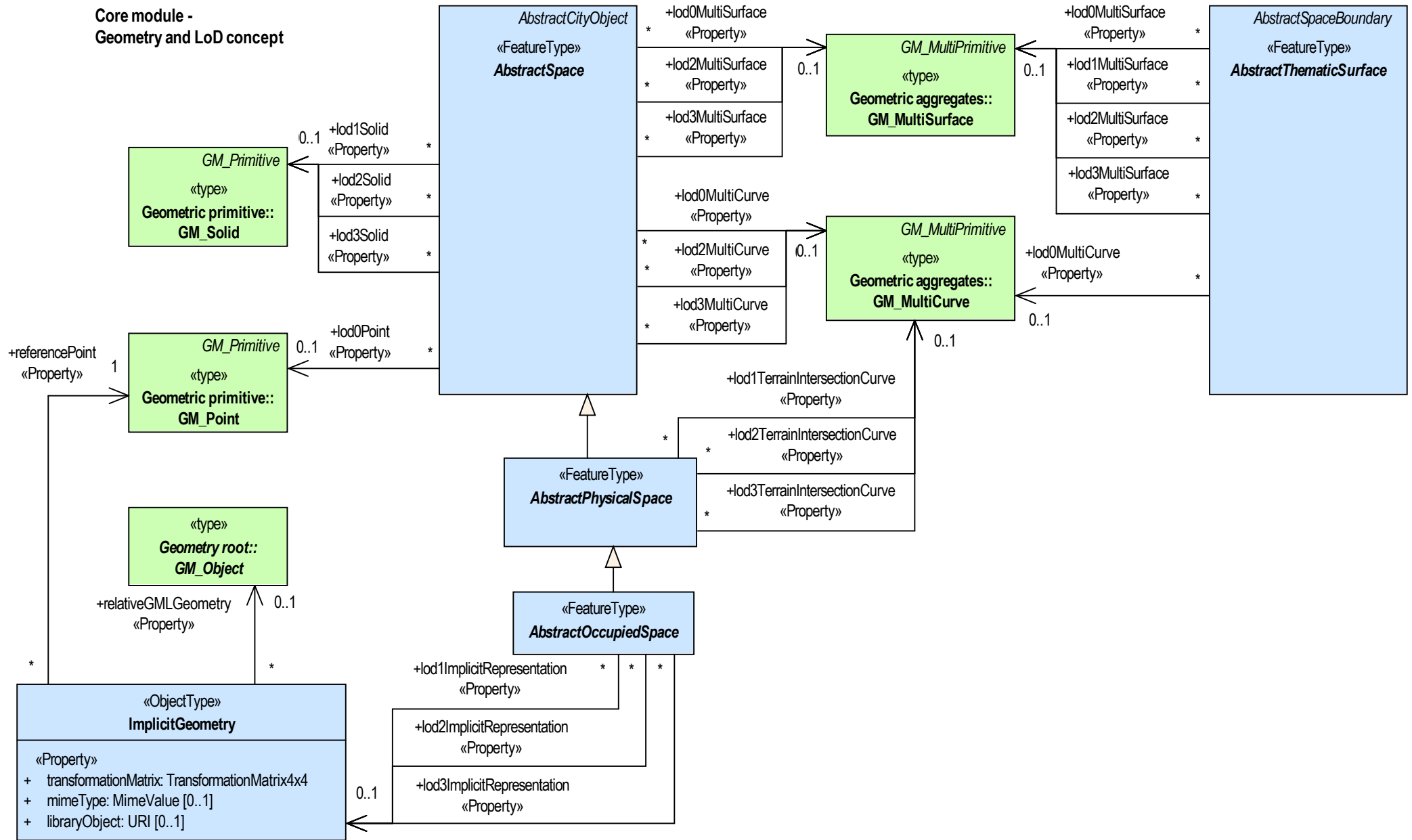




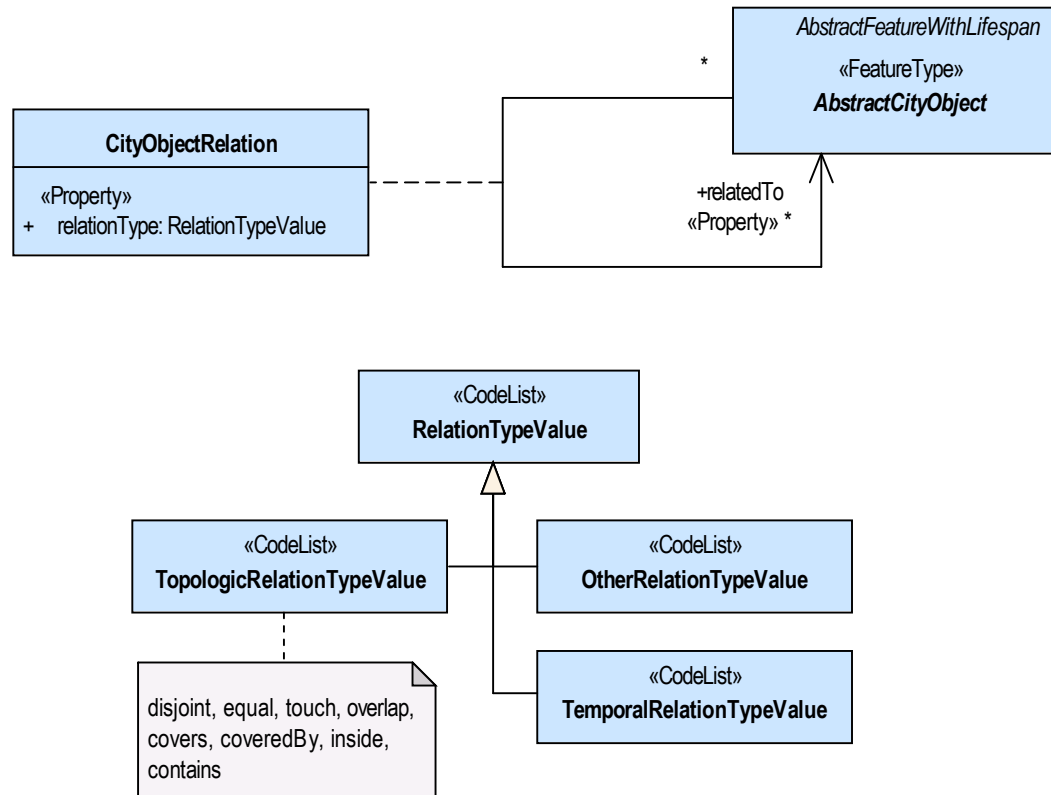
Core module - Space concepts



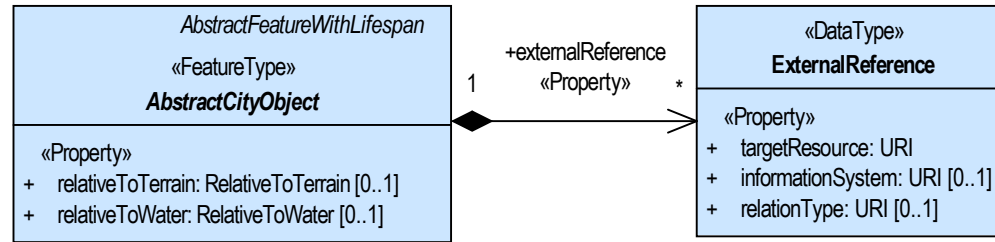
**Core module -
Geometry and LoD concept**



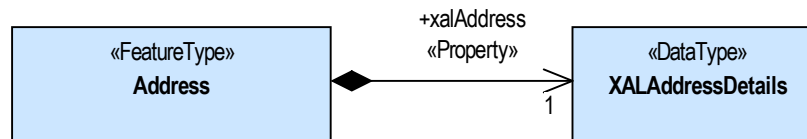
Core module - City object relations



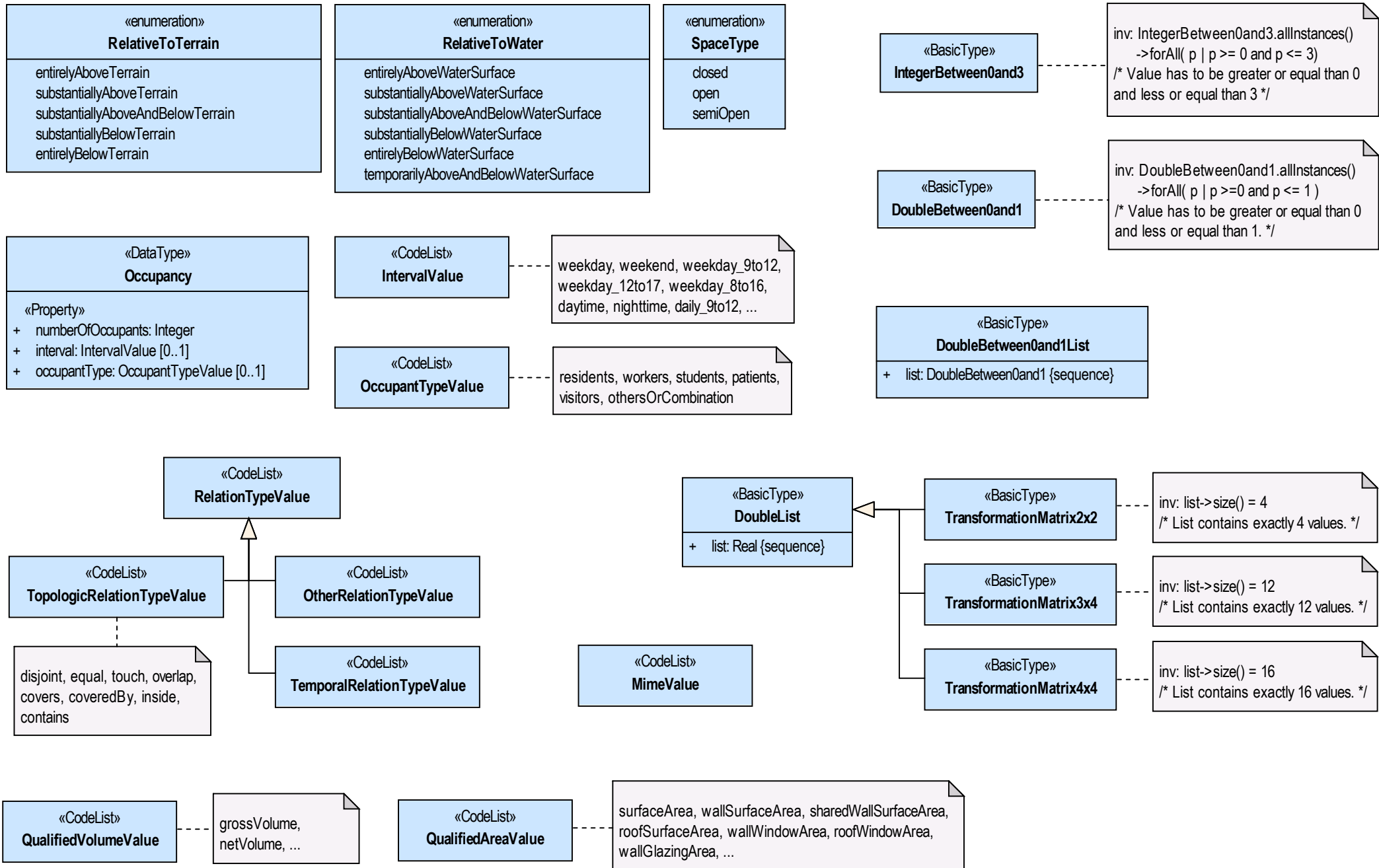
Core module - Miscellaneous

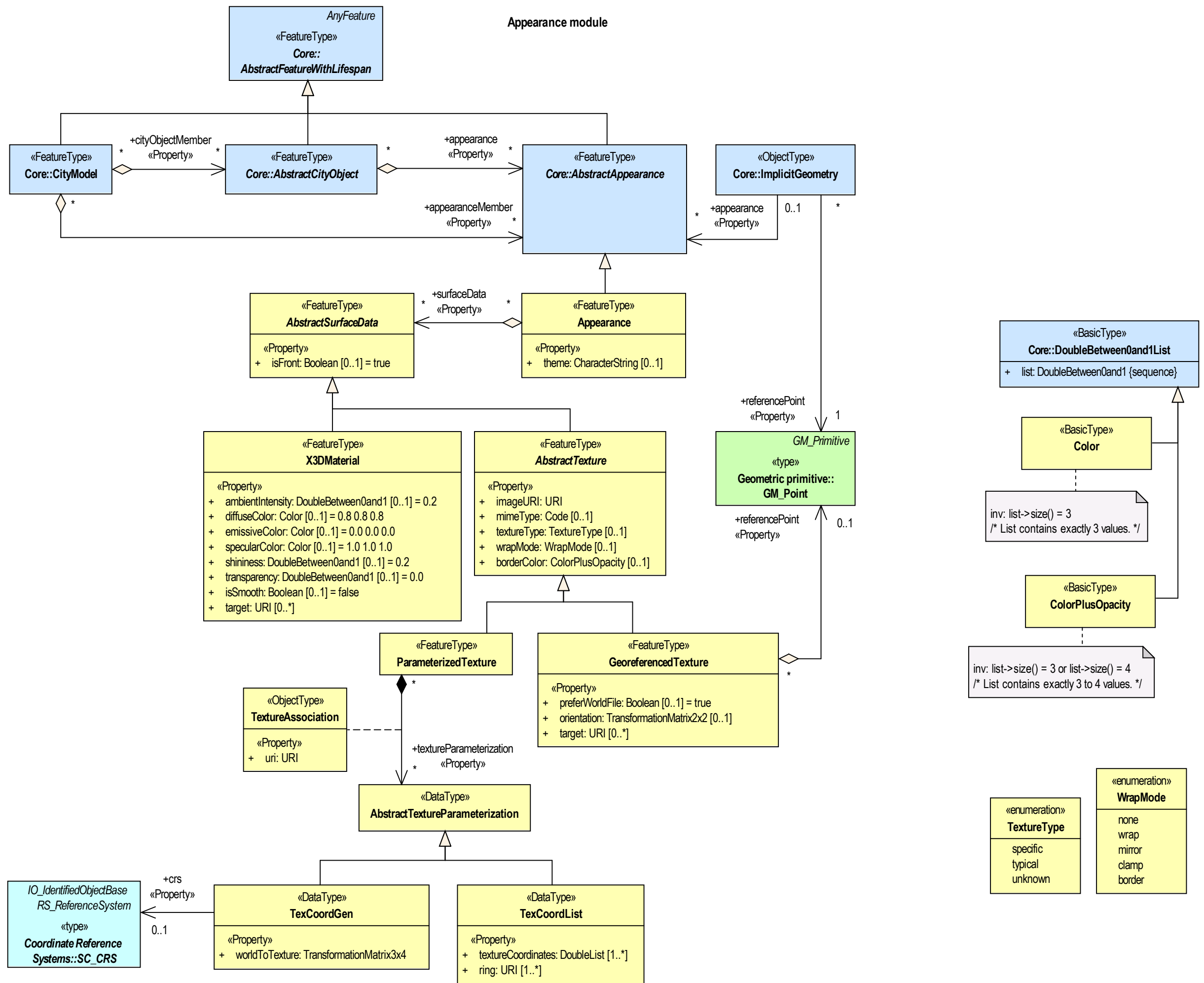


ExternalReference is now extended by an optional **relation Type** which can link to some external definition of the type of relation (e.g. the *sameAs* relation from OWL). Hence, **ExternalReferences** can now be used to express relations similar to RDF.

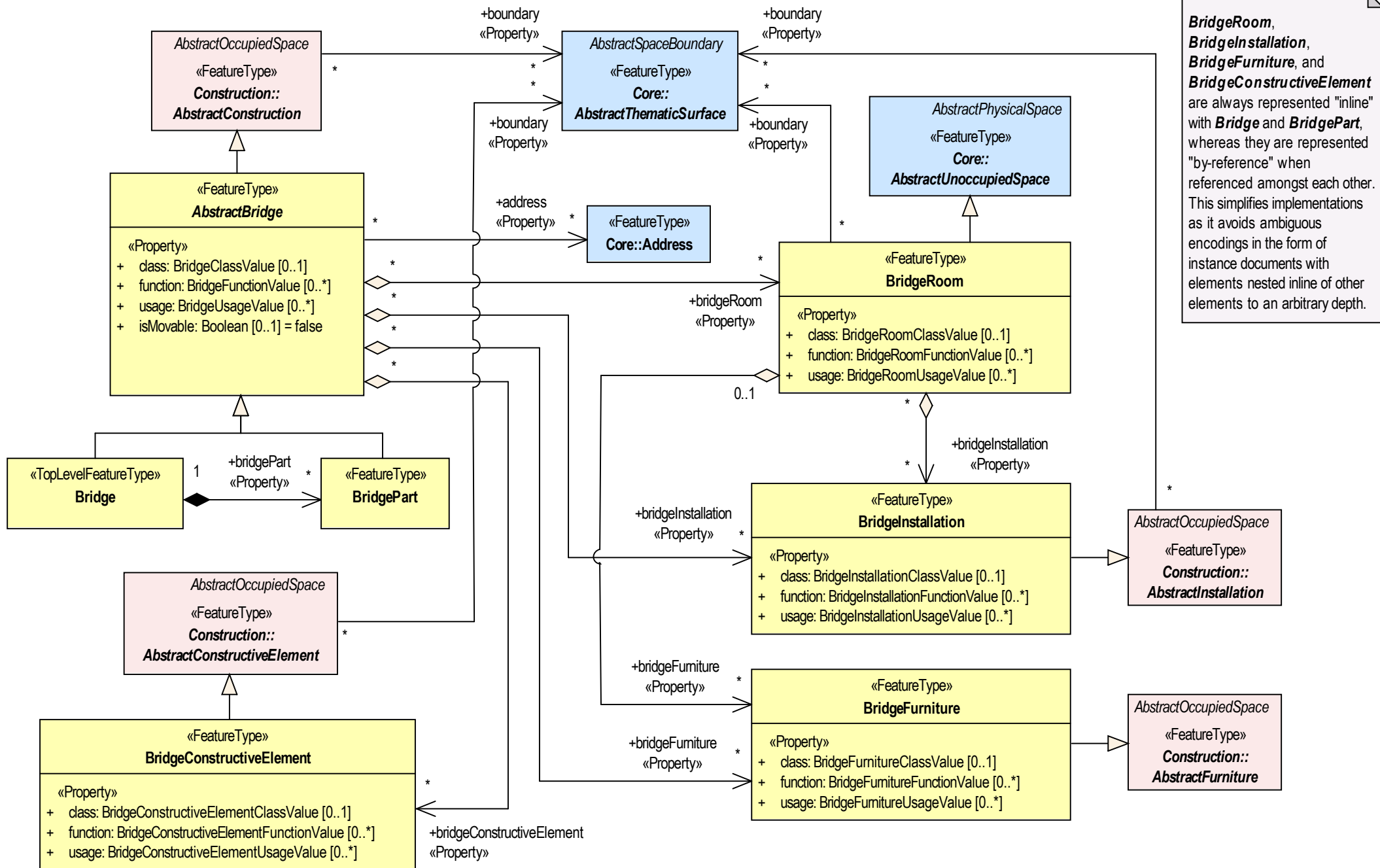


Core module - Basic Types, Enumerations, and Code lists

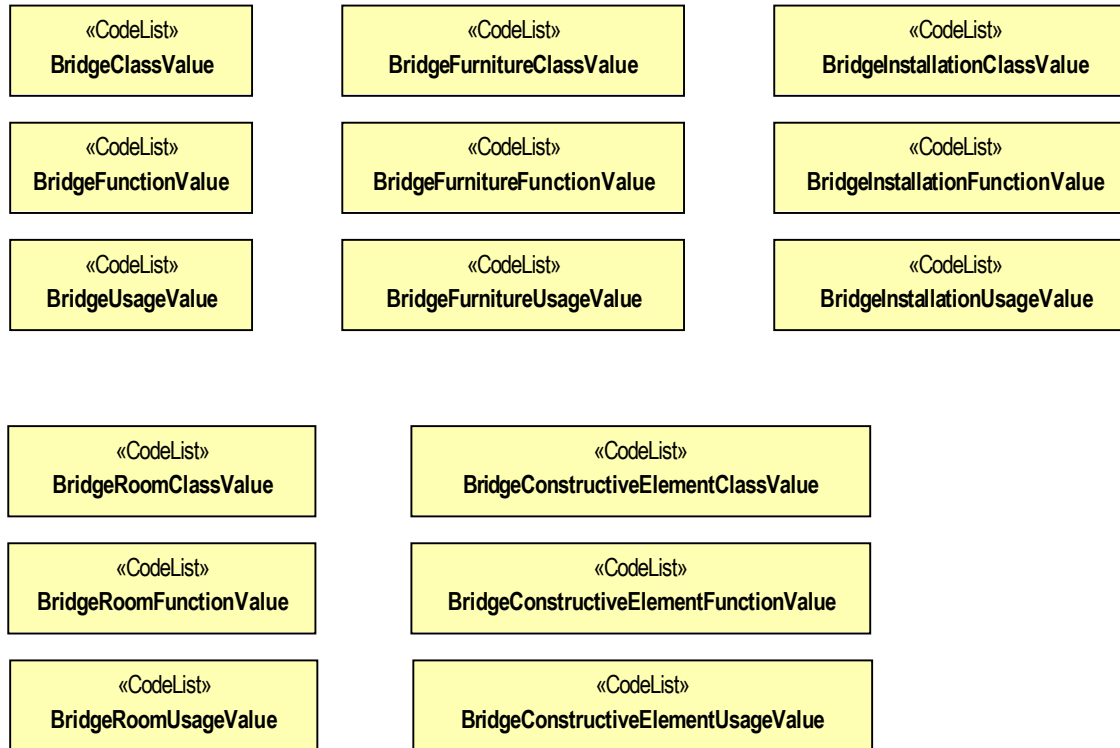




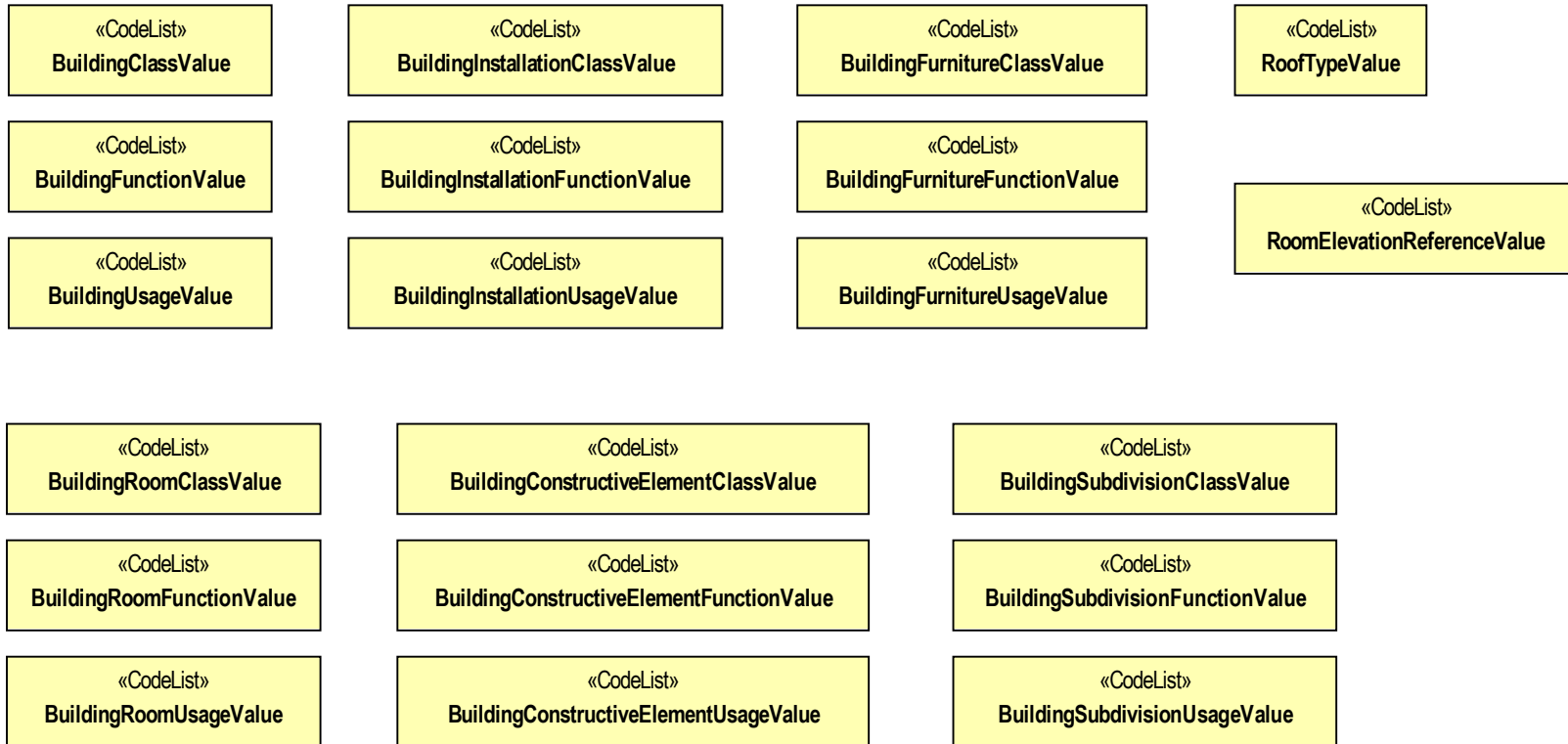
Bridge module



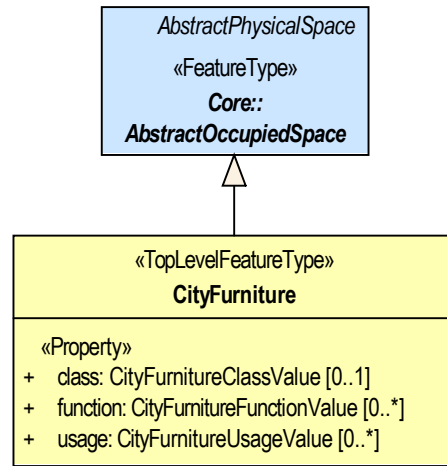
Bridge module - Code lists



Building module - Code lists



CityFurniture module



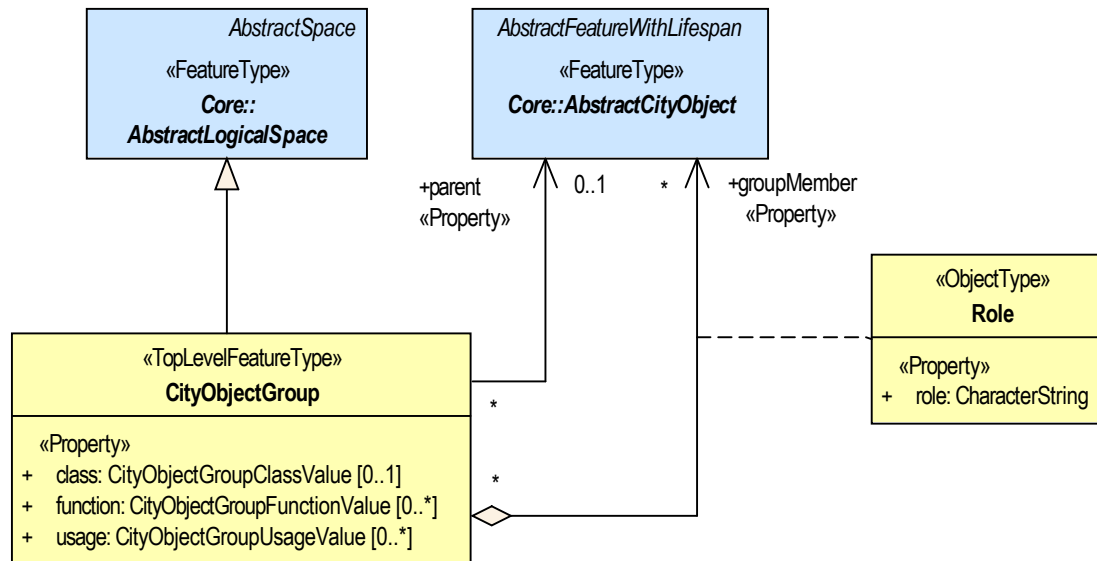
CityFurniture module - Code lists

«CodeList»
CityFurnitureClassValue

«CodeList»
CityFurnitureFunctionValue

«CodeList»
CityFurnitureUsageValue

CityObjectGroup module



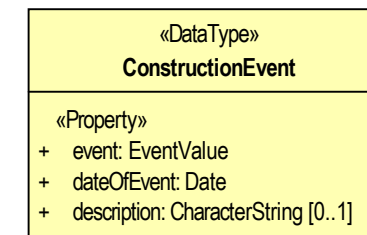
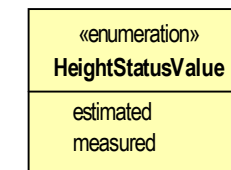
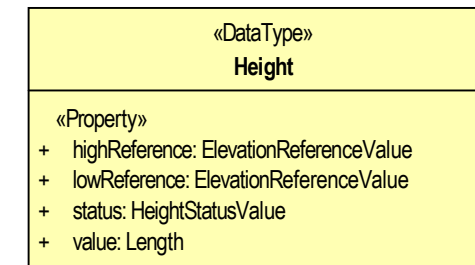
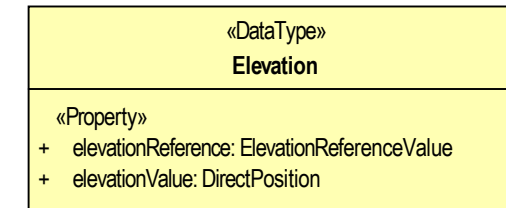
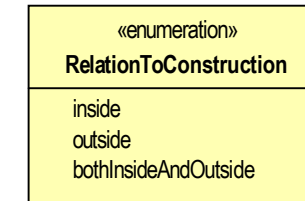
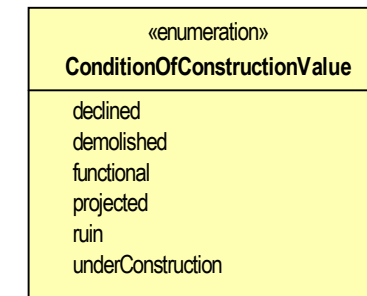
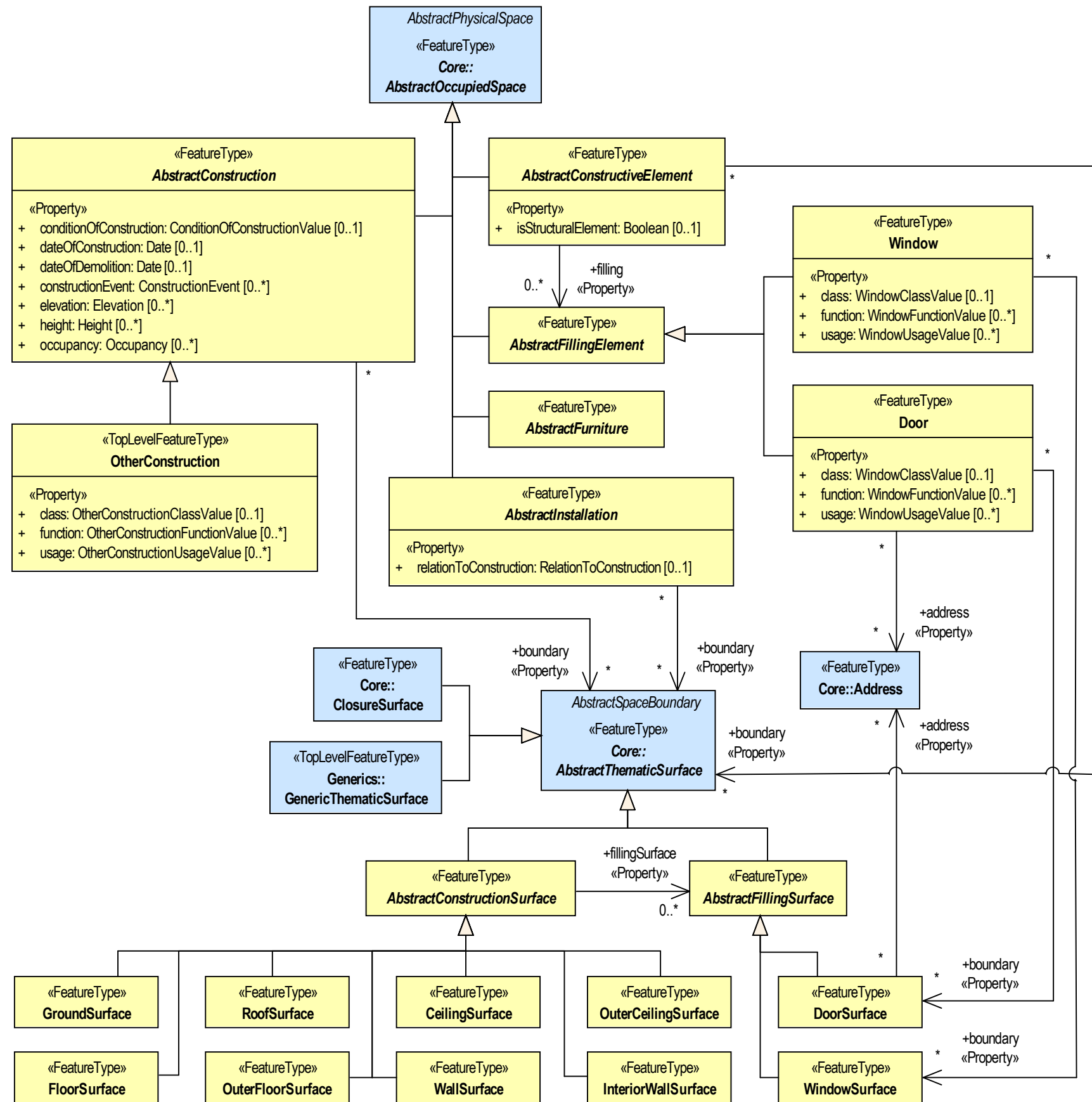
CityObjectGroup module - Code lists

«CodeList»
CityObjectGroupClassValue

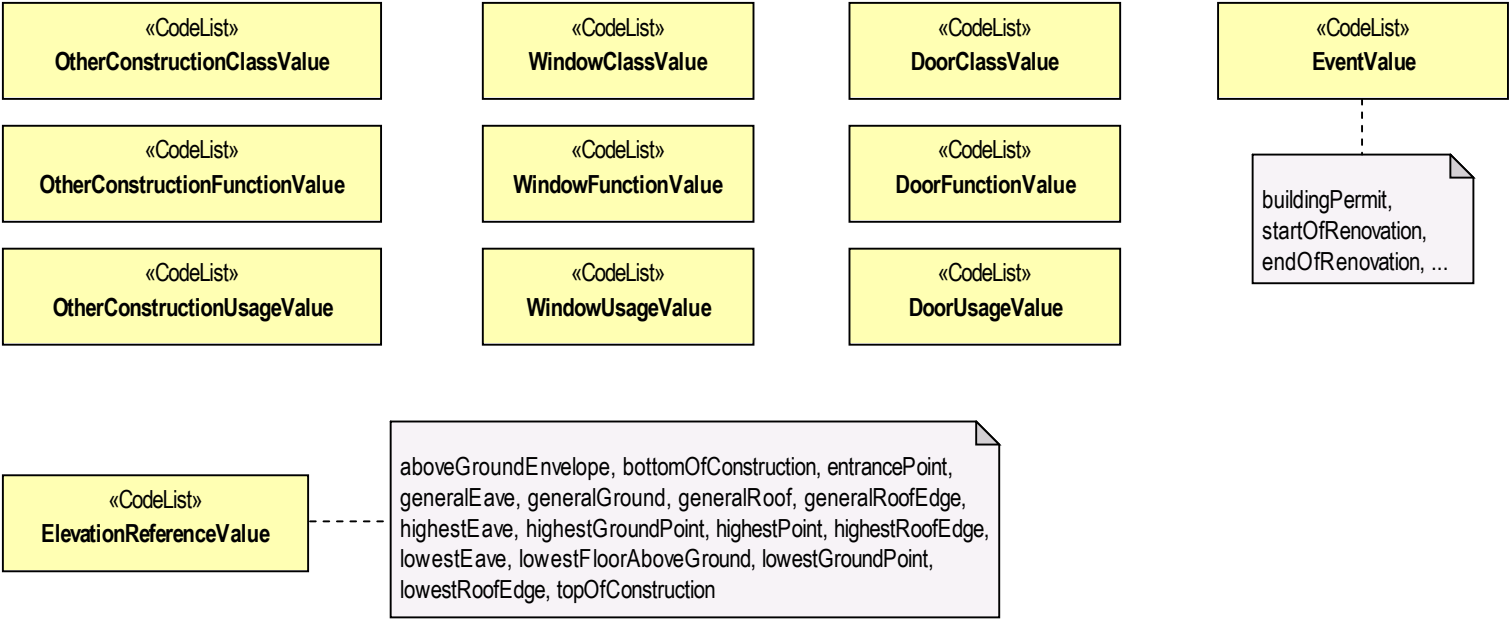
«CodeList»
CityObjectGroupFunctionValue

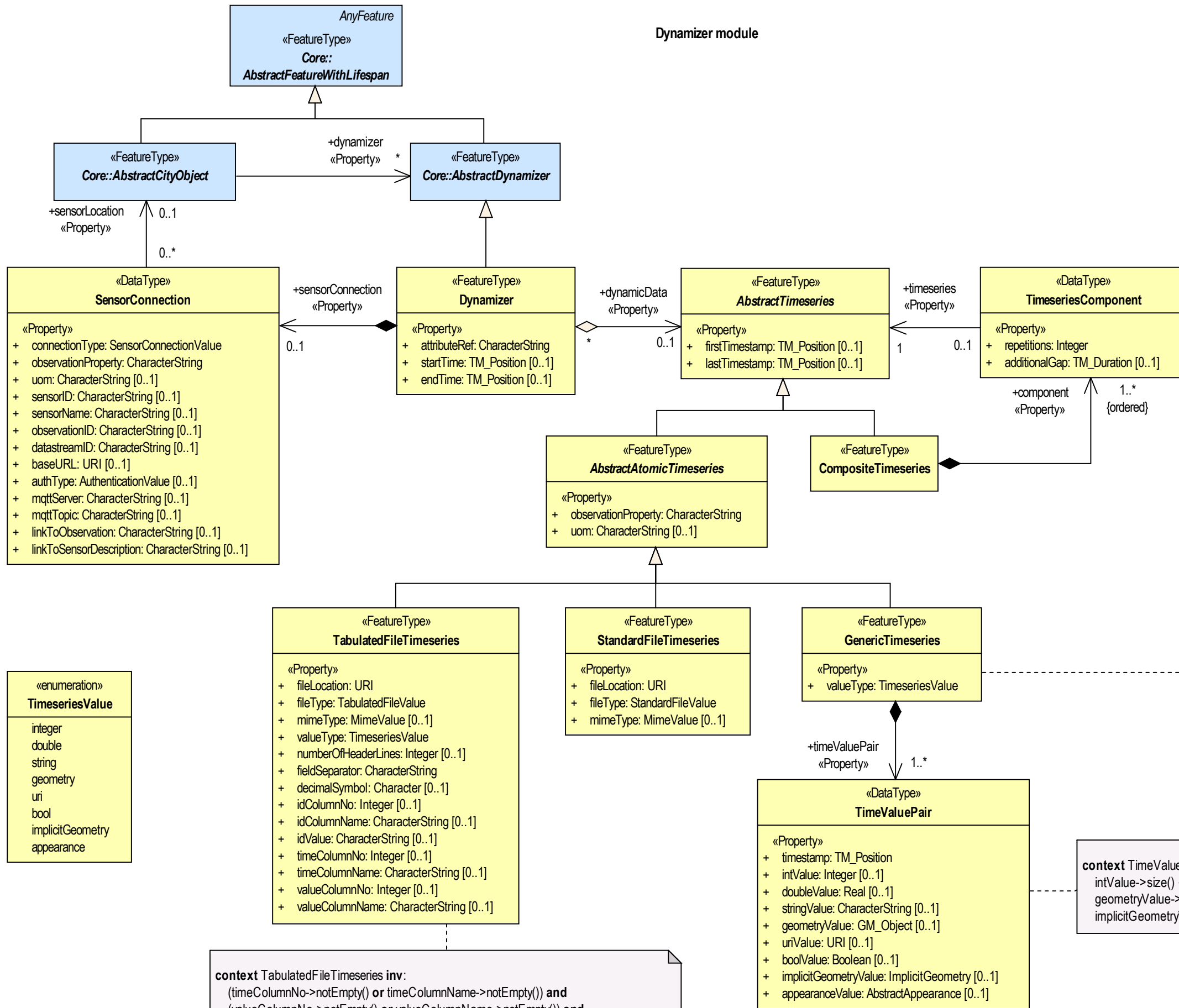
«CodeList»
CityObjectGroupUsageValue

Construction module



Construction module - Code lists





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
endif

```

context TimeValuePair inv:

```

intValue->size() + doubleValue->size() + stringValue->size() +
geometryValue->size() + uriValue->size() + boolValue->size() +
implicitGeometryValue->size() + appearanceValue->size() = 1

```

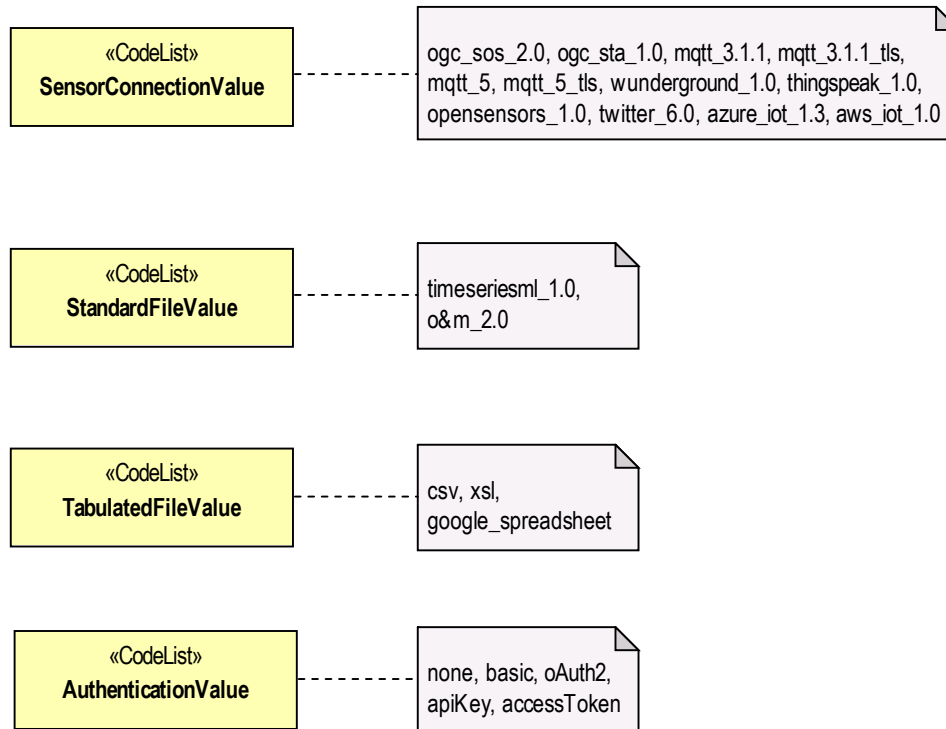
context TabulatedFileTimeseries inv:

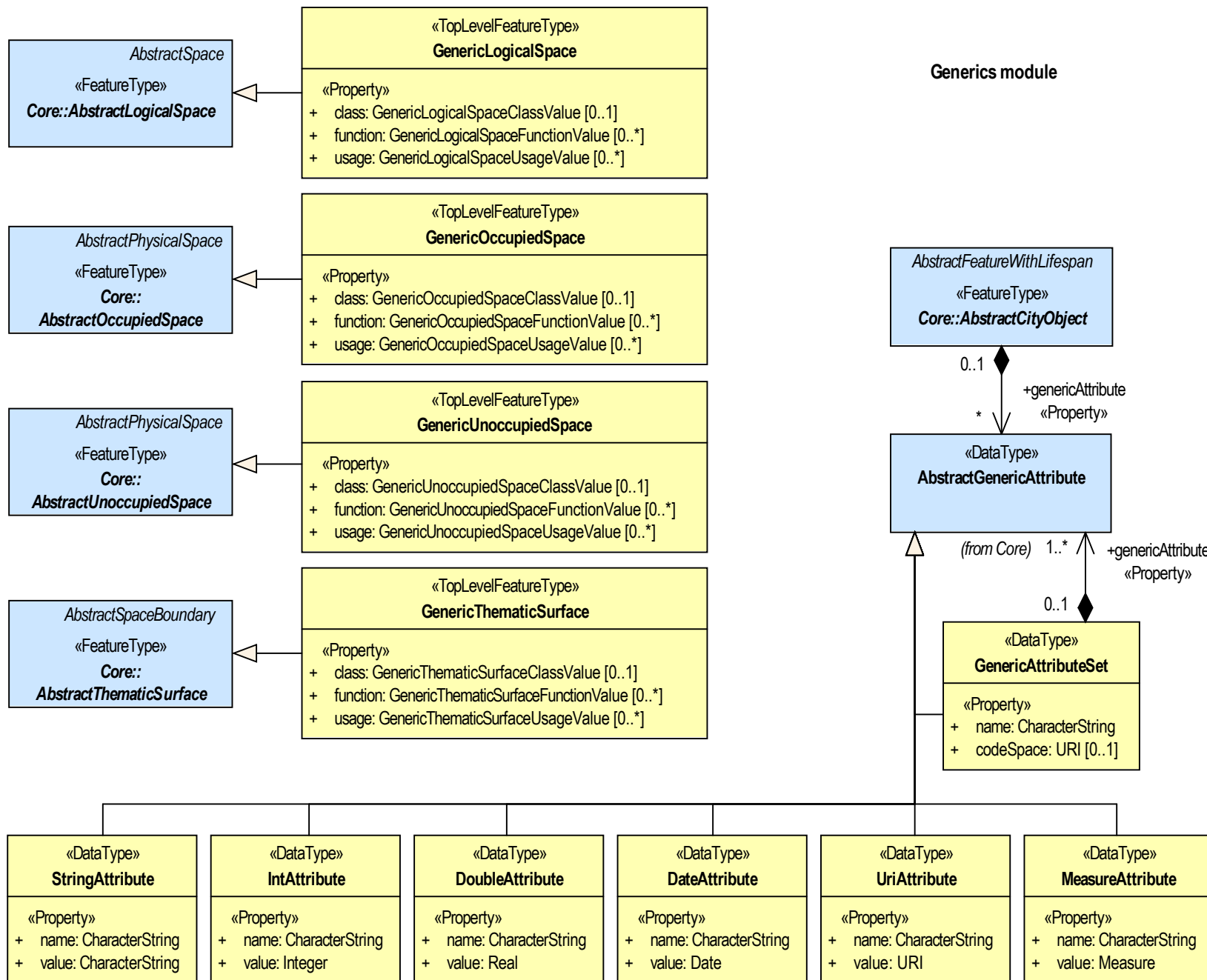
```

(timeColumnNo->notEmpty() or timeColumnName->notEmpty()) and
(valueColumnNo->notEmpty() or valueColumnName->notEmpty()) and
(idValue->notEmpty() implies idColumnNo->notEmpty() or idColumnName->notEmpty())

```

Dynamizer module - Code lists





Generics module - Code lists

«CodeList»
GenericLogicalSpaceClassValue

«CodeList»
GenericOccupiedSpaceClassValue

«CodeList»
GenericUnoccupiedSpaceClassValue

«CodeList»
GenericThematicSurfaceClassValue

«CodeList»
GenericLogicalSpaceFunctionValue

«CodeList»
GenericOccupiedSpaceFunctionValue

«CodeList»
GenericUnoccupiedSpaceFunctionValue

«CodeList»
GenericThematicSurfaceFunctionValue

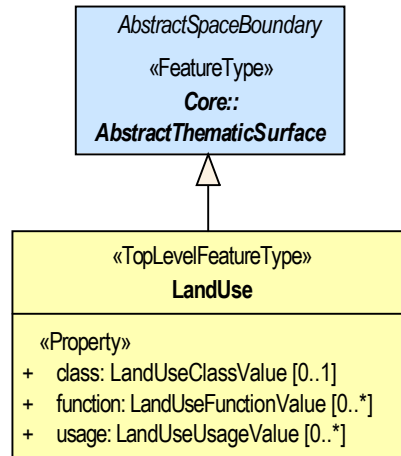
«CodeList»
GenericLogicalSpaceUsageValue

«CodeList»
GenericOccupiedSpaceUsageValue

«CodeList»
GenericUnoccupiedSpaceUsageValue

«CodeList»
GenericThematicSurfaceUsageValue

LandUse module



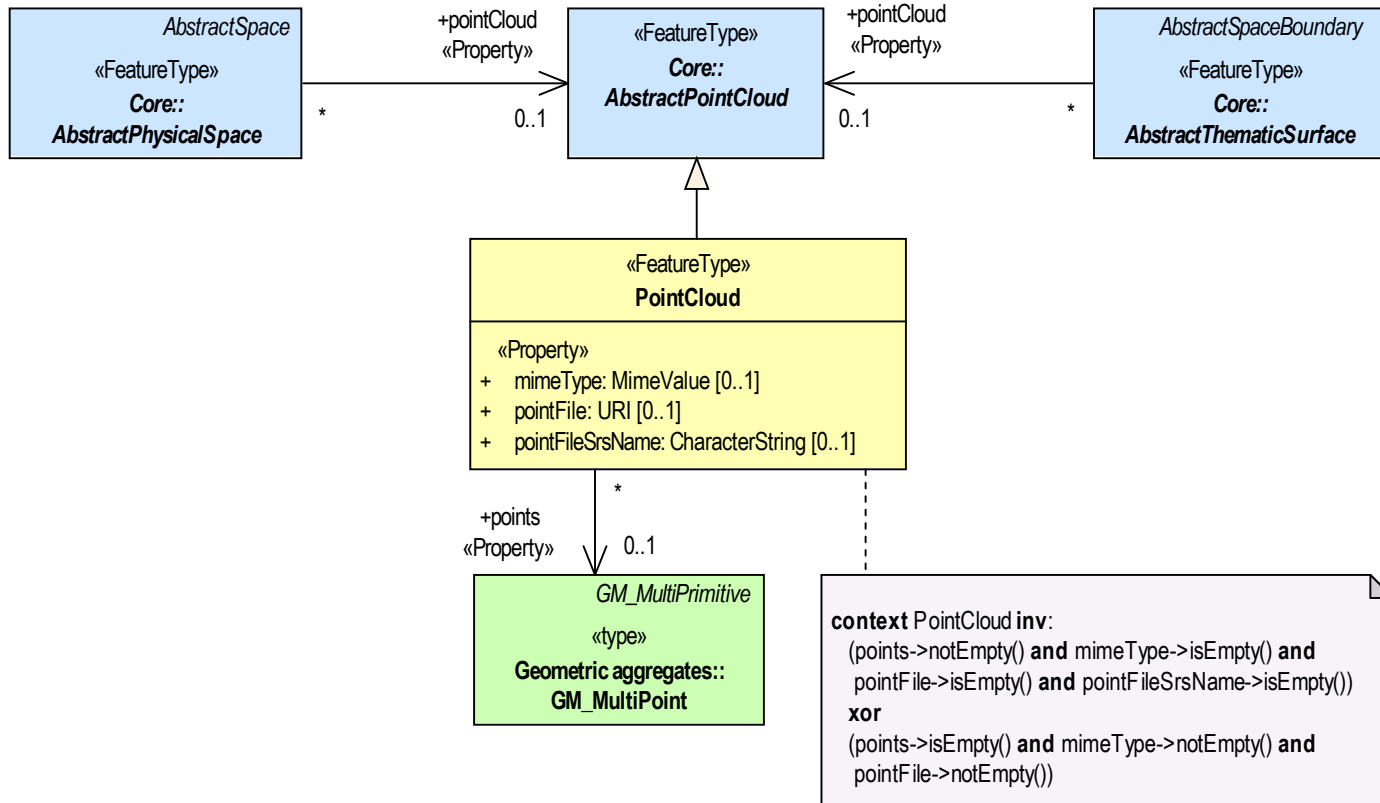
LandUse module - Code lists

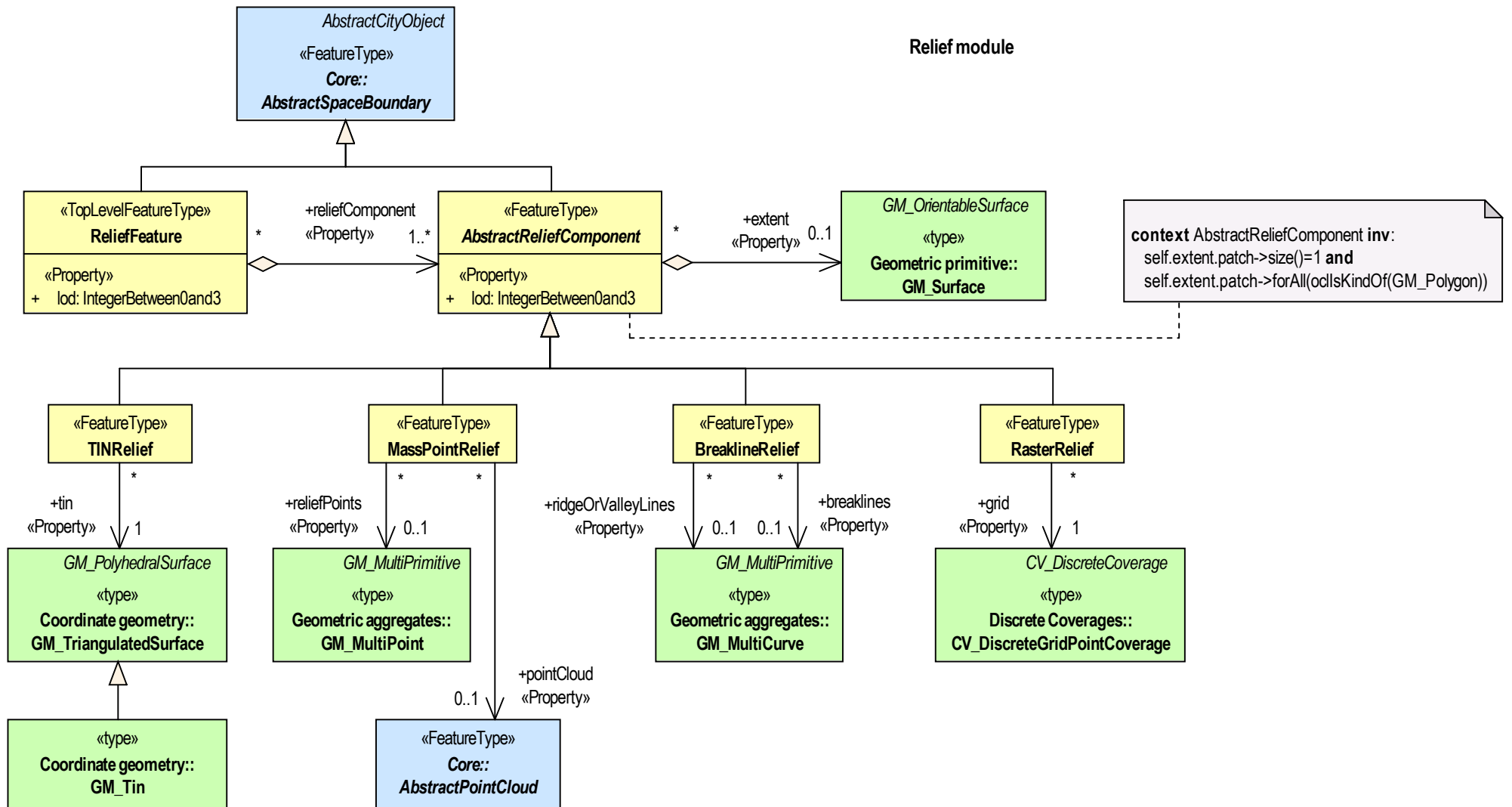
«CodeList»
LandUseClassValue

«CodeList»
LandUseFunctionValue

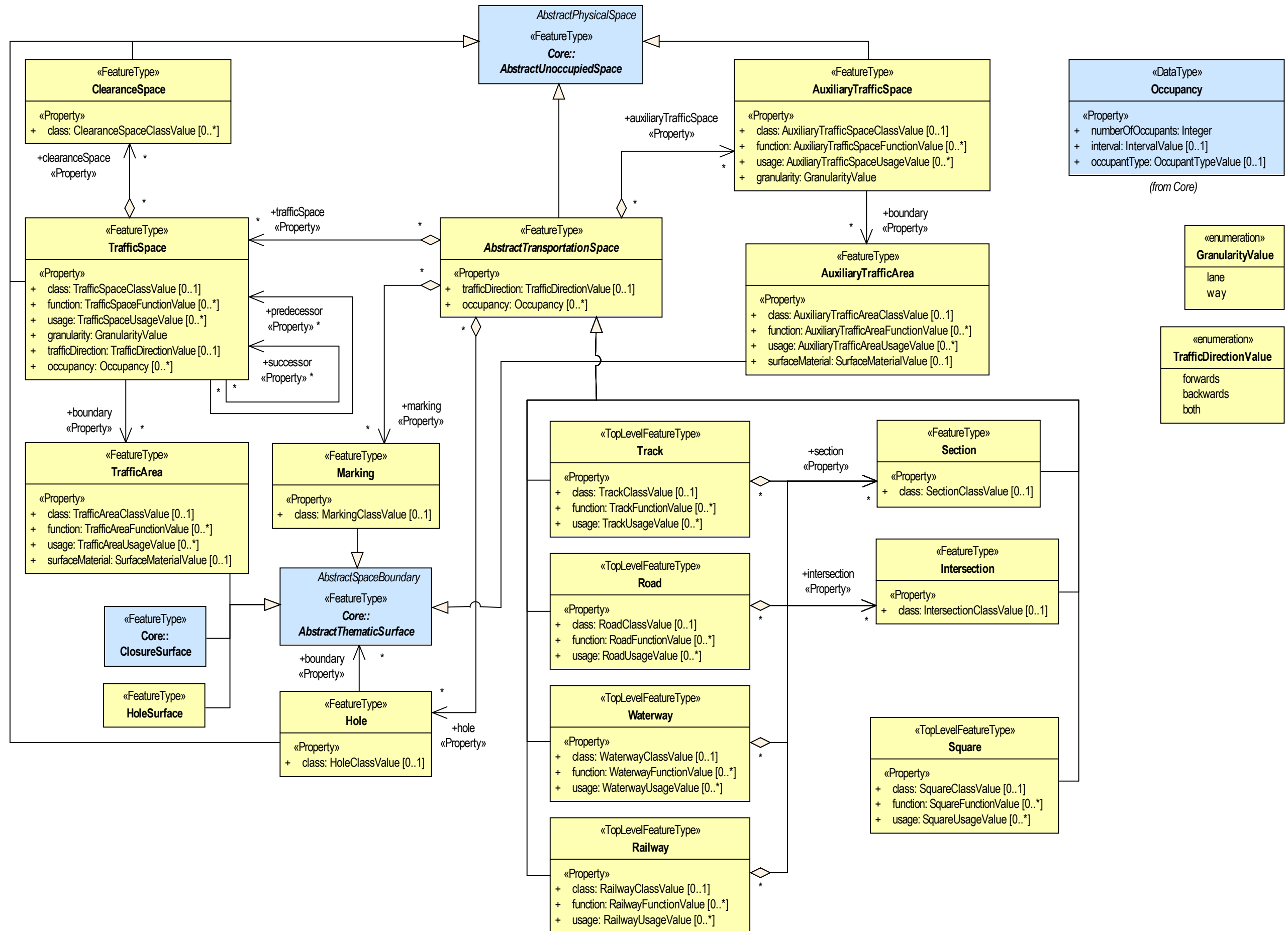
«CodeList»
LandUseUsageValue

PointCloud module

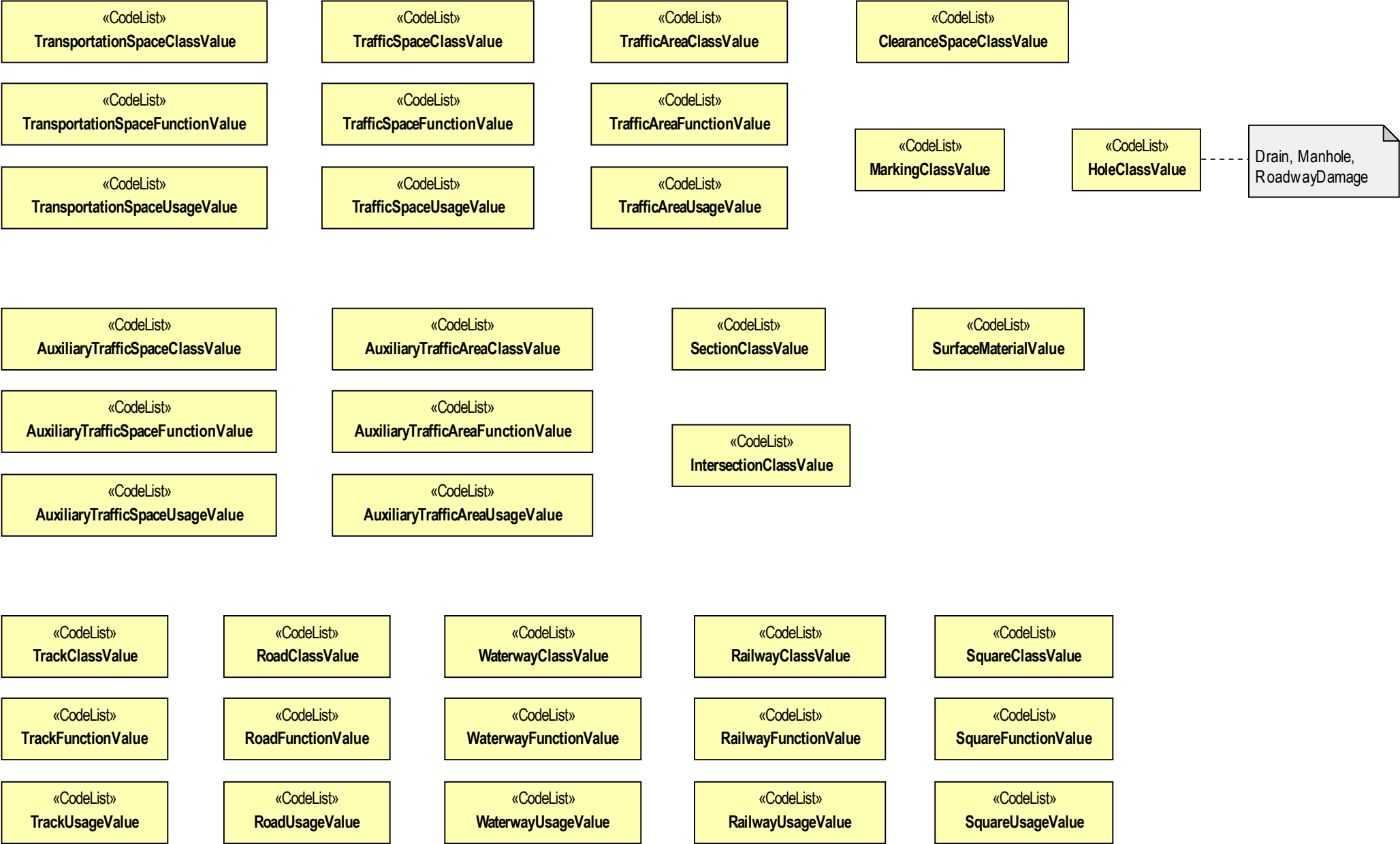




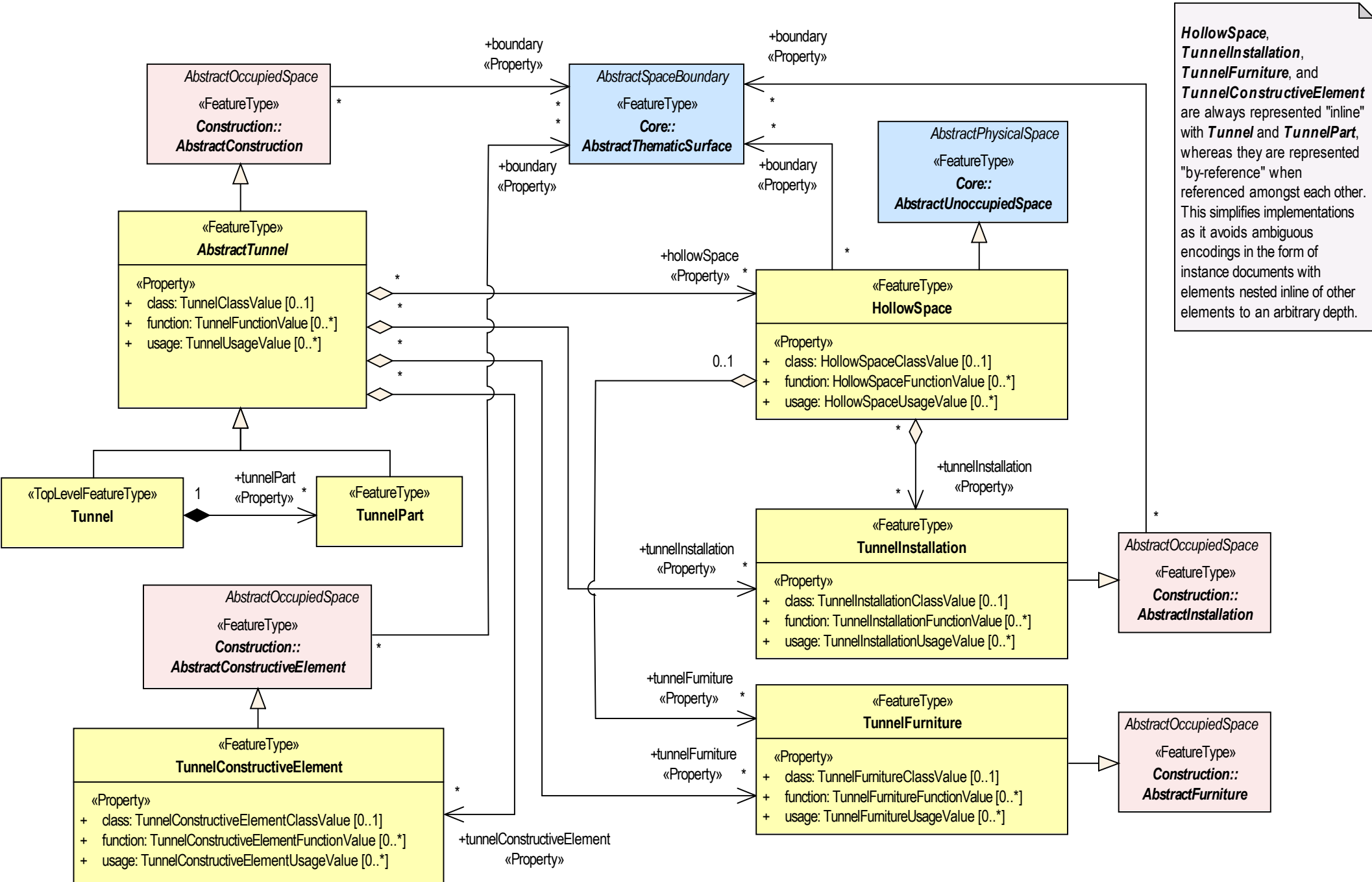
Transportation module



Transportation module - Code lists



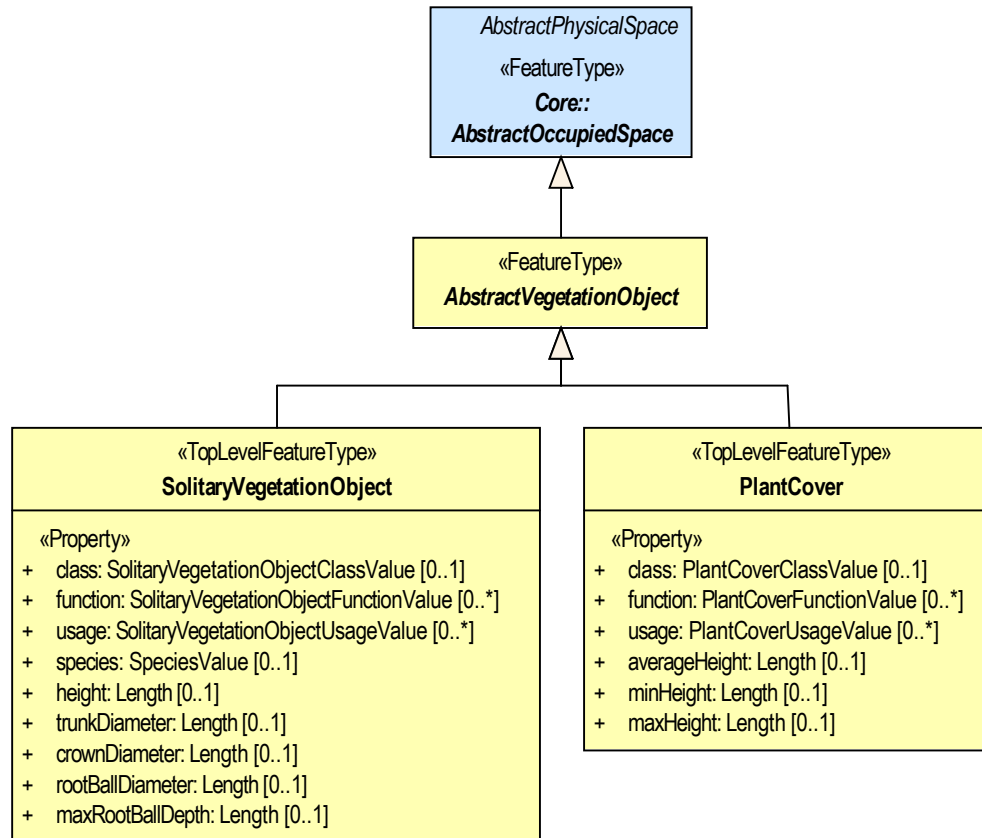
Tunnel module



Tunnel module - Code lists

«CodeList» TunnelClassValue	«CodeList» TunnelInstallationClassValue	«CodeList» TunnelFurnitureClassValue
«CodeList» TunnelFunctionValue	«CodeList» TunnelInstallationFunctionValue	«CodeList» TunnelFurnitureFunctionValue
«CodeList» TunnelUsageValue	«CodeList» TunnelInstallationUsageValue	«CodeList» TunnelFurnitureUsageValue
«CodeList» HollowSpaceClassValue	«CodeList» TunnelConstructiveElementClassValue	
«CodeList» HollowSpaceFunctionValue	«CodeList» TunnelConstructiveElementFunctionValue	
«CodeList» HollowSpaceUsageValue	«CodeList» TunnelConstructiveElementUsageValue	

Vegetation module



Vegetation module - Code lists

«CodeList»
SolitaryVegetationObjectClassValue

«CodeList»
SolitaryVegetationObjectFunctionValue

«CodeList»
SolitaryVegetationObjectUsageValue

«CodeList»
PlantCoverClassValue

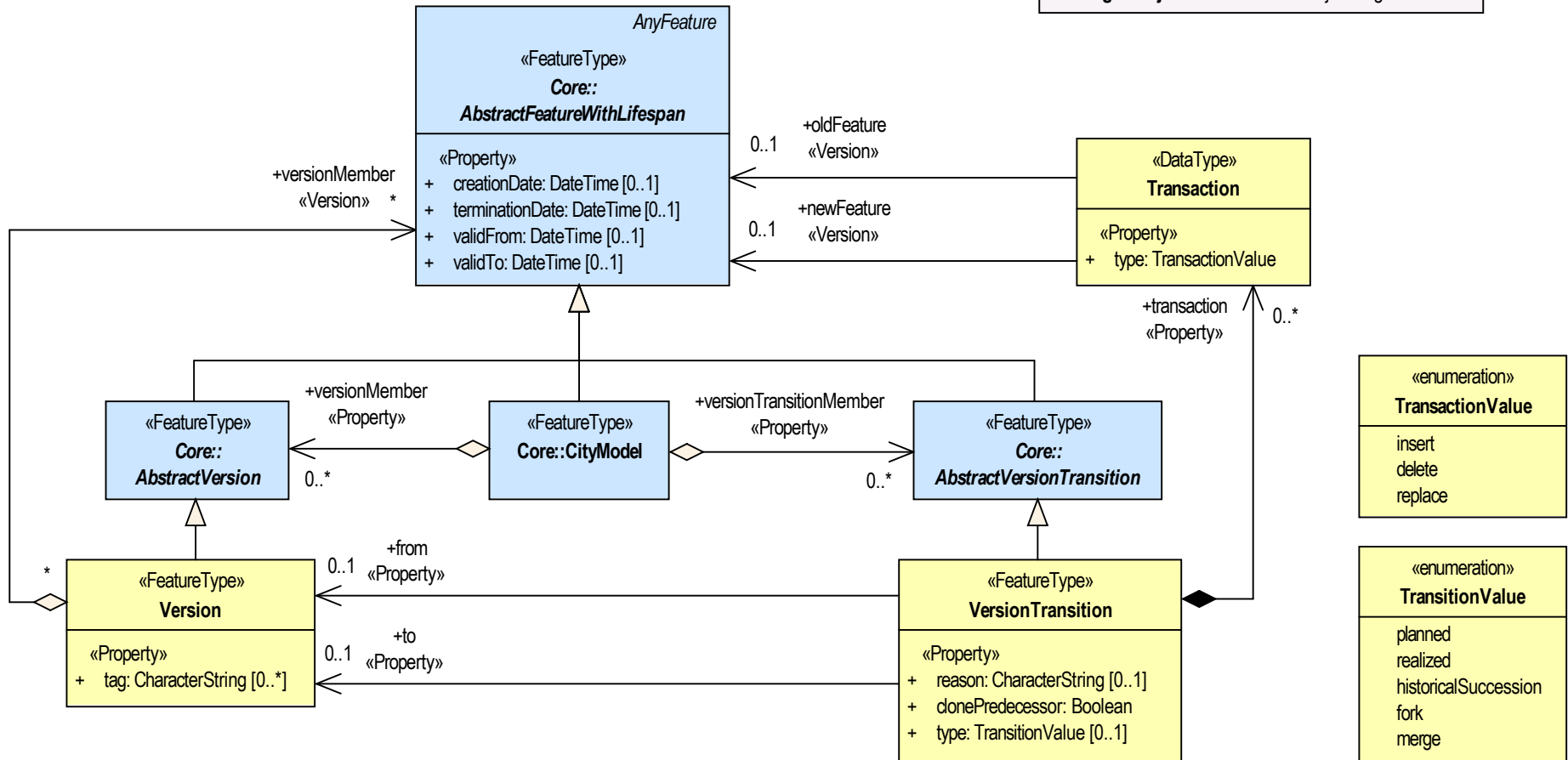
«CodeList»
PlantCoverFunctionValue

«CodeList»
PlantCoverUsageValue

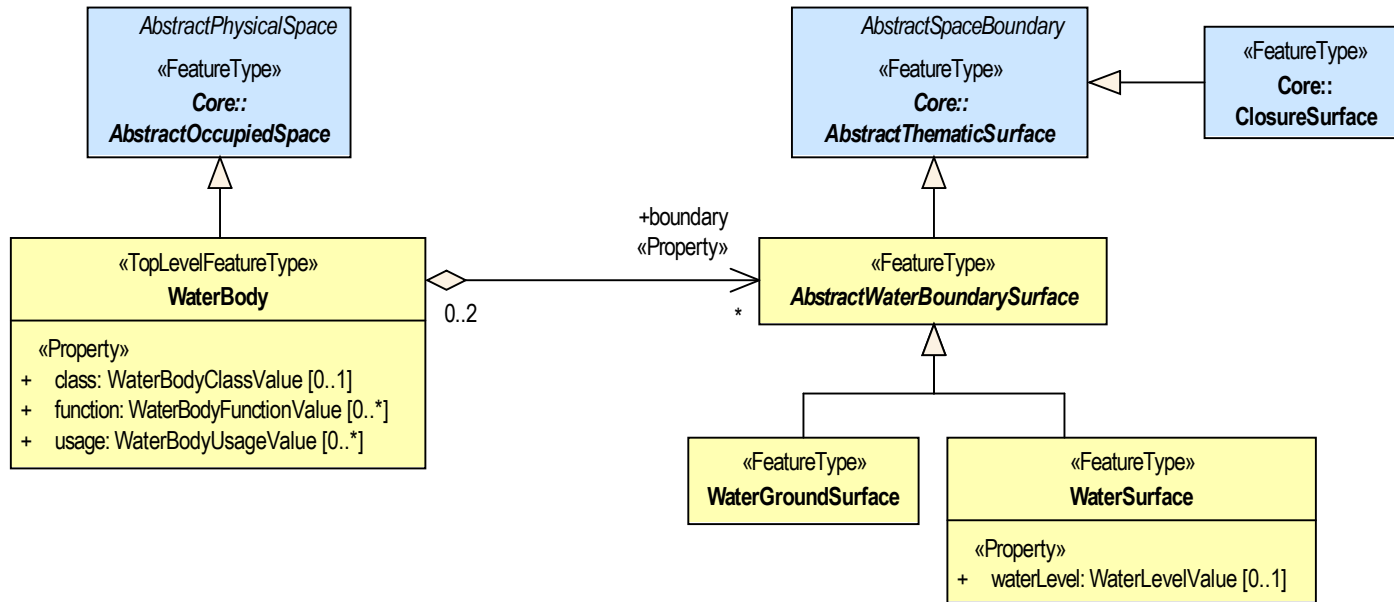
«CodeList»
SpeciesValue

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

«CodeList»
WaterBodyClassValue

«CodeList»
WaterLevelValue

«CodeList»
WaterBodyFunctionValue

«CodeList»
WaterBodyUsageValue