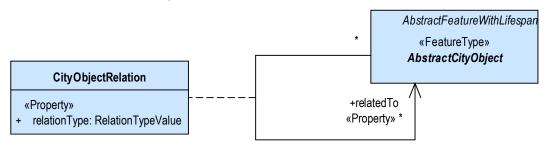
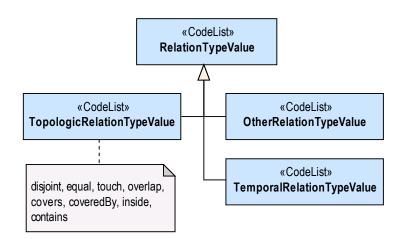
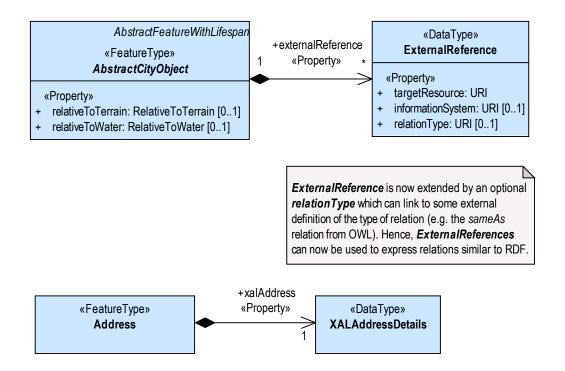


## Core module - City object relations

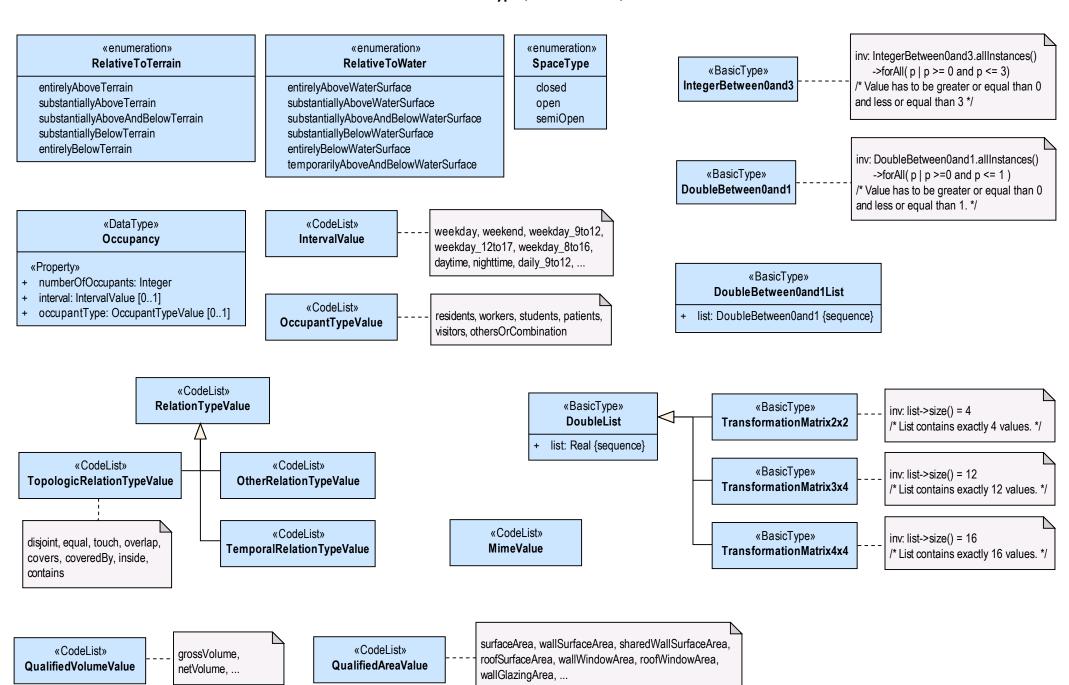


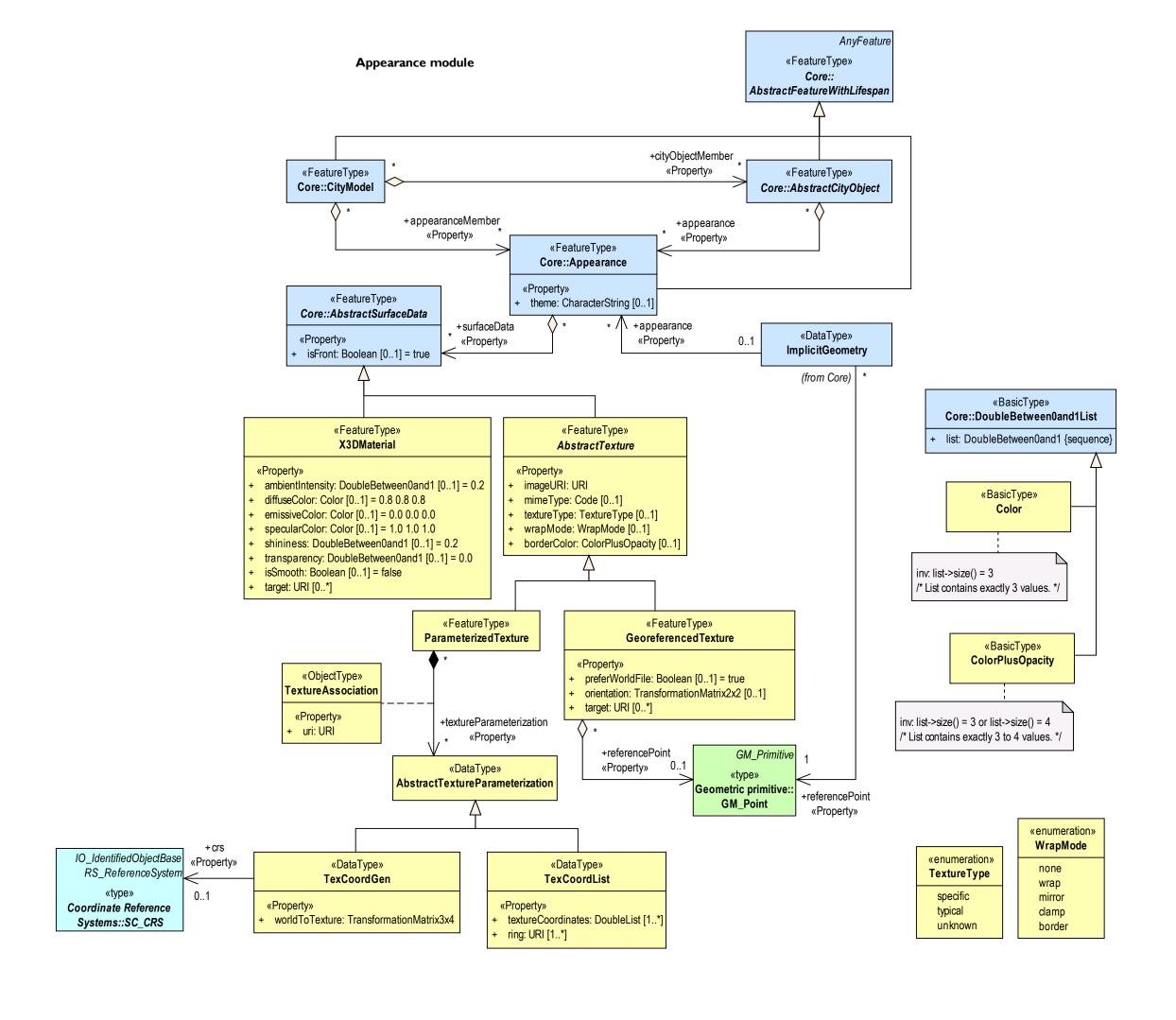


#### Core module - Miscellaneous

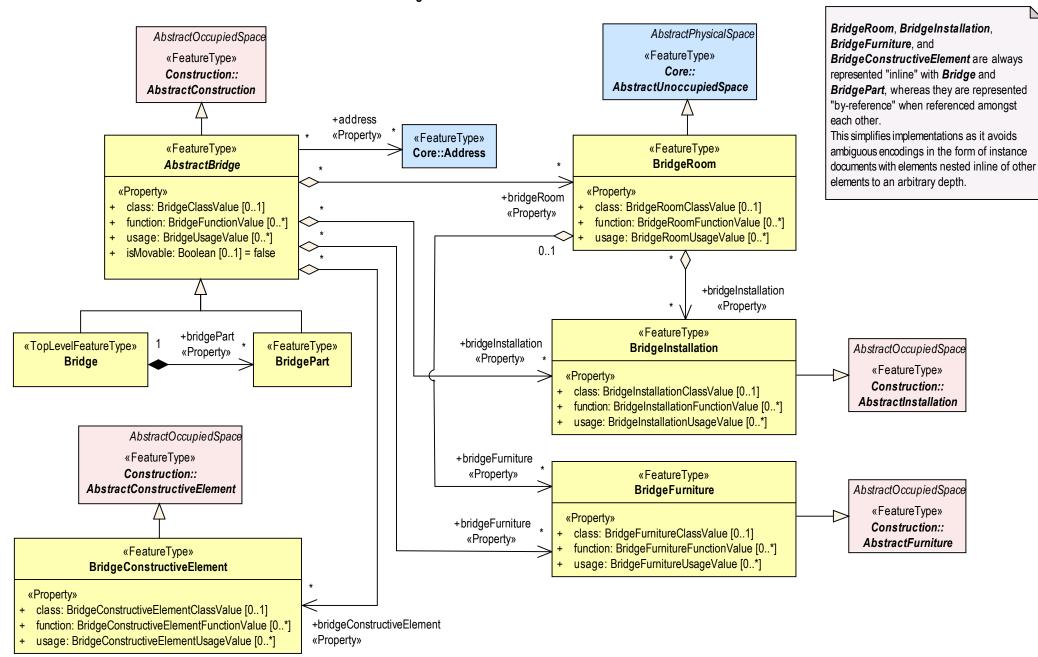


#### Core module - Basic Types, Enumerations, and Code lists





#### Bridge module



## Bridge module - Code lists

«CodeList»

BridgeClassValue

«CodeList»
BridgeFurnitureClassValue

«CodeList»
BridgeInstallationClassValue

«CodeList»

BridgeFunctionValue

«CodeList»
BridgeFurnitureFunctionValue

«CodeList»
BridgeInstallationFunctionValue

«CodeList»
BridgeUsageValue

«CodeList»
BridgeFurnitureUsageValue

«CodeList»
BridgeInstallationUsageValue

«CodeList»
BridgeRoomClassValue

«CodeList»

BridgeConstructiveElementClassValue

«CodeList»

BridgeRoomFunctionValue

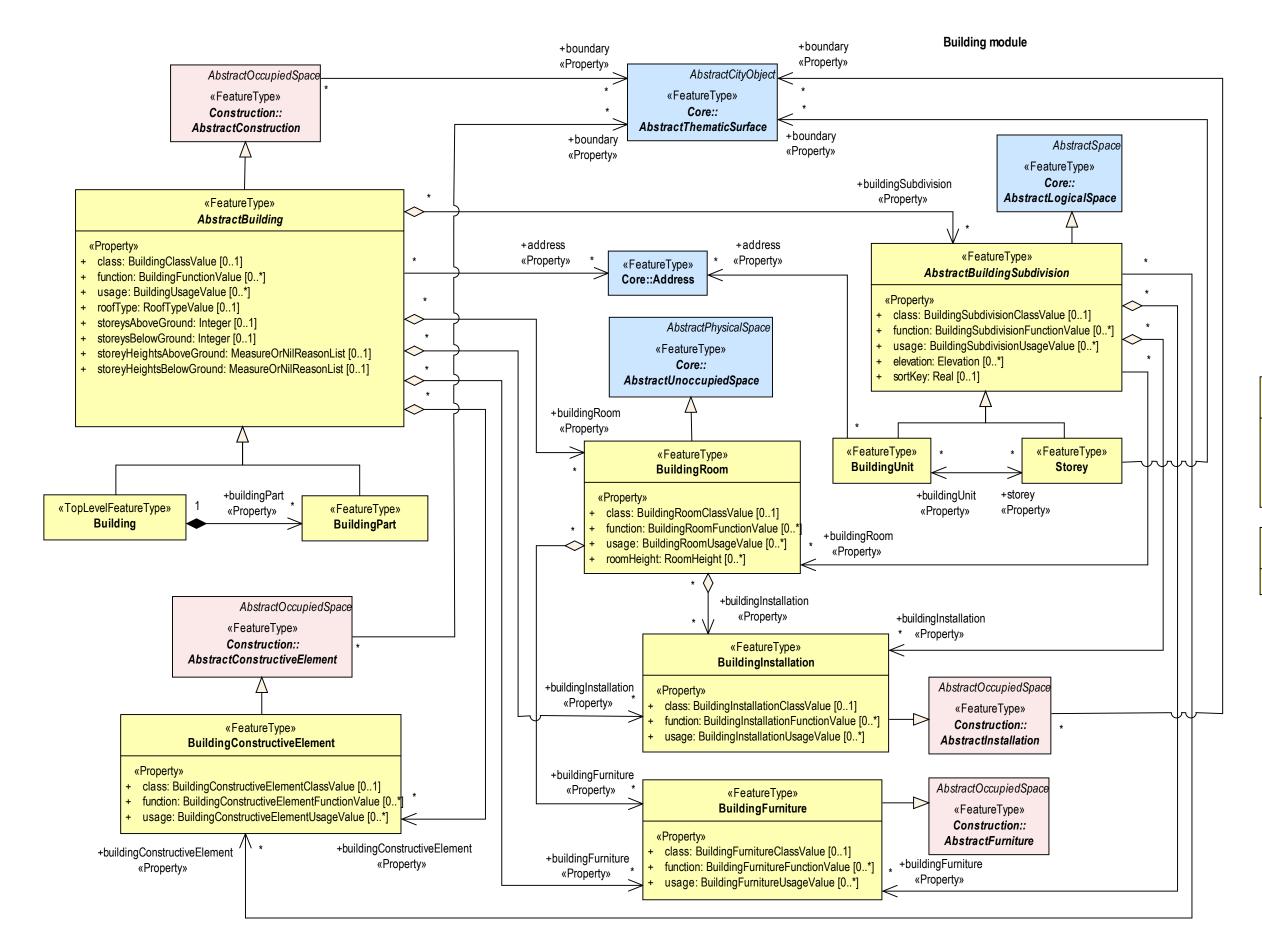
 $\label{eq:codeList} $$\ensuremath{\mathsf{WCodeList}}$ \\ \textbf{BridgeConstructiveElementFunctionValue}$ 

«CodeList»

BridgeRoomUsageValue

«CodeList»

BridgeConstructiveElementUsageValue



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**

«CodeList»

BuildingClassValue

«CodeList»

BuildingInstallationClassValue

«CodeList»

BuildingFurnitureClassValue

«CodeList»
RoofTypeValue

«CodeList»

BuildingFunctionValue

«CodeList»

BuildingInstallationFunctionValue

«CodeList»

BuildingFurnitureFunctionValue

«CodeList»

BuildingUsageValue

«CodeList»

BuildingInstallationUsageValue

«CodeList»

BuildingFurnitureUsageValue

«CodeList»

BuildingRoomClassValue

«CodeList»

BuildingConstructiveElementClassValue

«CodeList»

BuildingSubdivisionClassValue

«CodeList»

BuildingRoomFunctionValue

 $\label{eq:CodeList} $$ \mbox{BuildingConstructiveElementFunctionValue} $$$ 

«CodeList» **BuildingSubdivisionFunctionValue** 

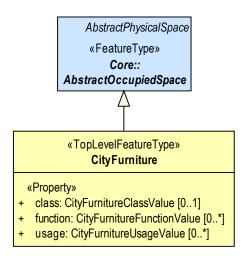
«CodeList»

BuildingRoomUsageValue

 $\label{eq:codeList} $$\operatorname{BuildingConstructiveElementUsageValue}$$ 

«CodeList» **BuildingSubdivisionUsageValue** 

# CityFurniture module



# CityFurniture module - Code lists

«CodeList»

CityFurnitureClassValue

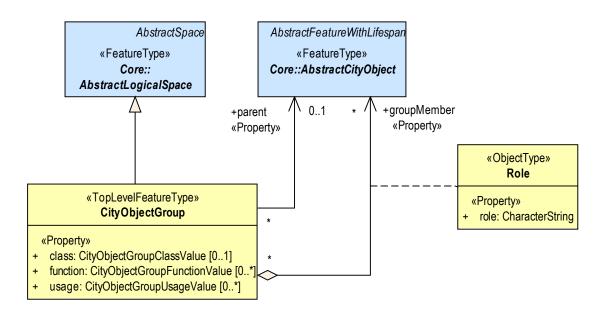
«CodeList»

CityFurnitureFunctionValue

«CodeList»

CityFurnitureUsageValue

## CityObjectGroup module



# CityObjectGroup module - Code lists

«CodeList»
CityObjectGroupClassValue

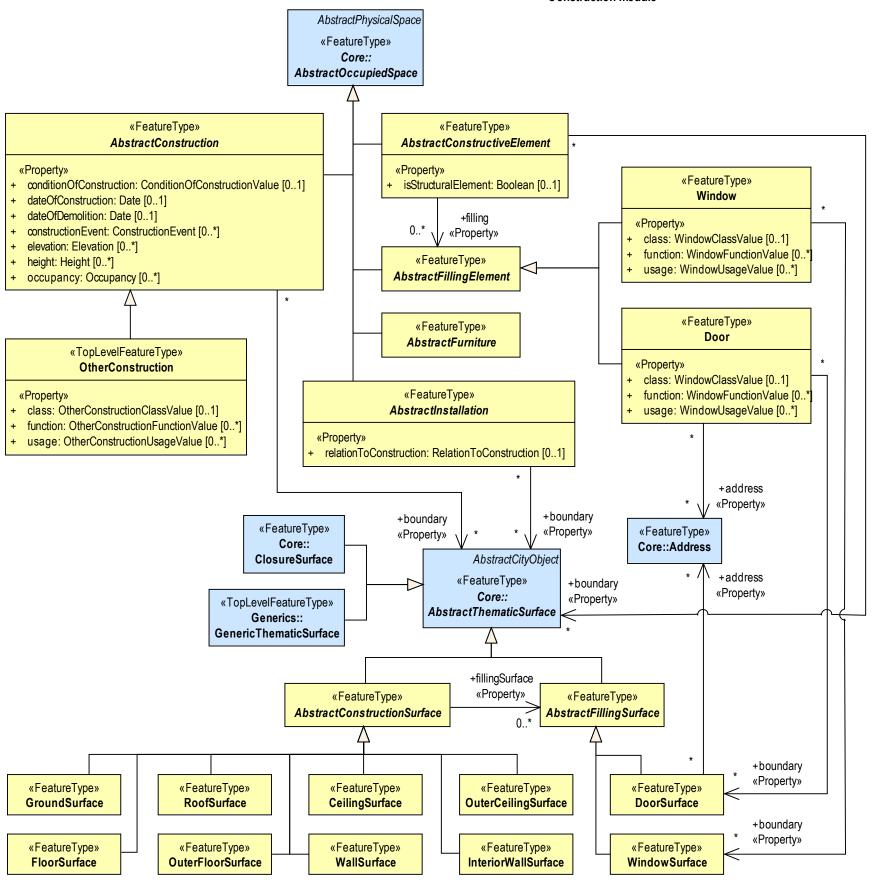
«CodeList»

CityObjectGroupFunctionValue

«CodeList»

CityObjectGroupUsageValue

#### Construction module



«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

# «enumeration» HeightStatusValue

estimated measured

Reference values for heights below ground will be added.

# «DataType» ConstructionEvent

- «Property»
- event: EventValue

IowestRoofEdge

topOfConstruction

- dateOfEvent: Date
- description: CharacterString [0..1]

## Construction module - Code lists

«CodeList»
OtherConstructionClassValue

«CodeList»
WindowClassValue

«CodeList»

DoorClassValue

«CodeList»
WindowFunctionValue

«CodeList»

DoorFunctionValue

 $\label{eq:codeList} $$\operatorname{\textbf{OtherConstructionFunctionValue}}$$ 

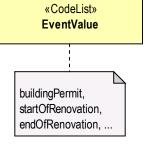
ndowrunctionvalue

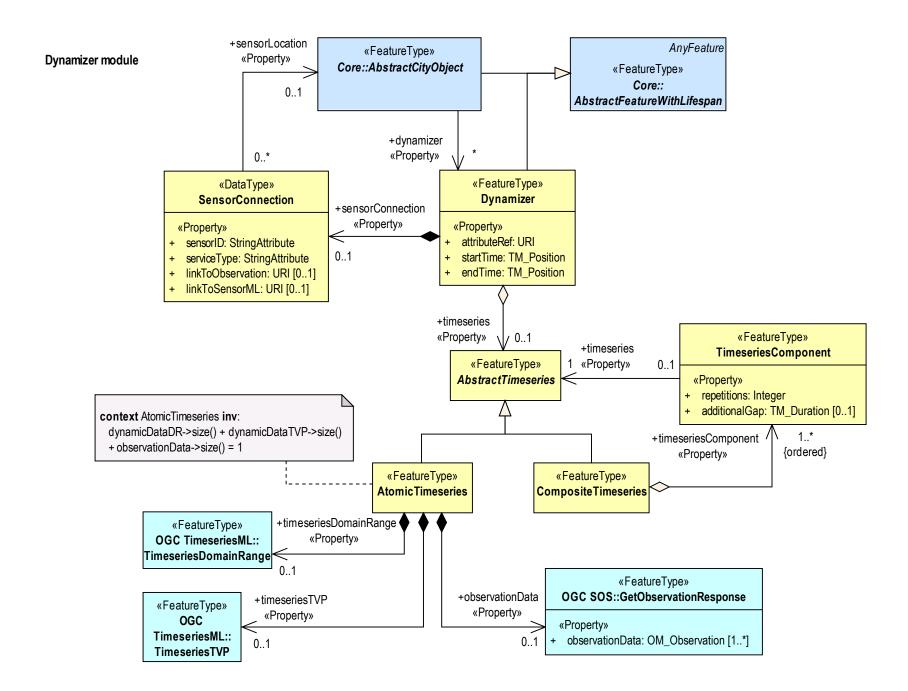
«CodeList»

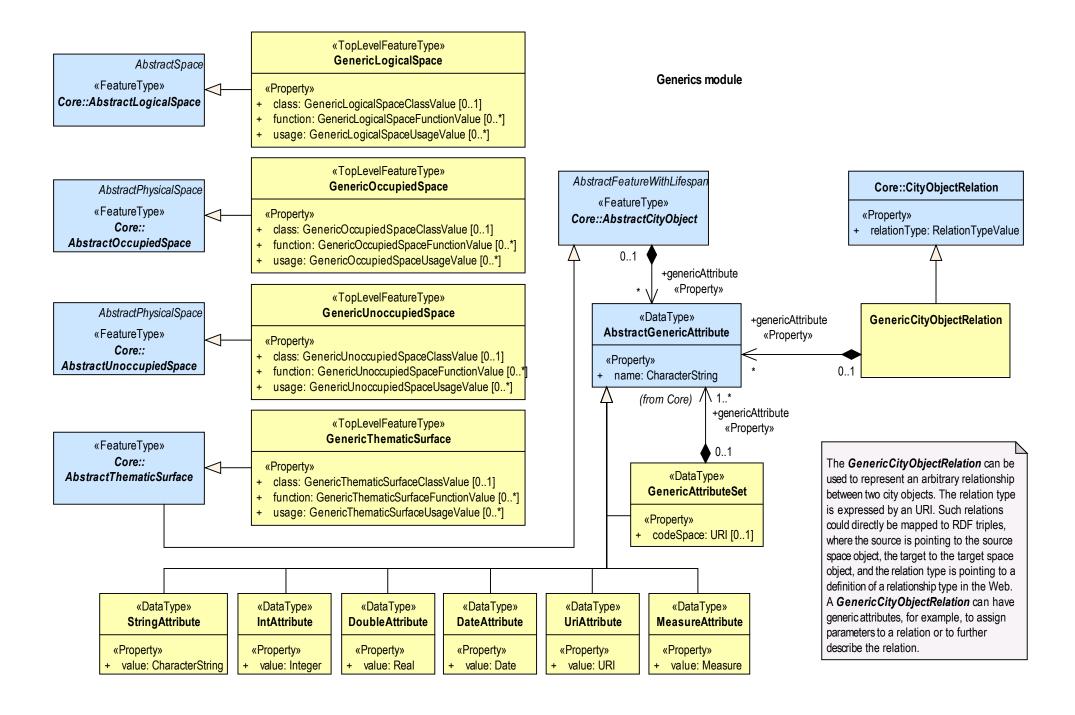
DoorUsageValue

«CodeList»
OtherConstructionUsageValue

«CodeList» WindowUsageValue







#### Generics module - Code lists

«CodeList»

GenericLogicalSpaceClassValue

«CodeList»

GenericOccupiedSpaceClassValue

«CodeList» «CodeList»

GenericUnoccupiedSpaceClassValue GenericThematicSurfaceClassValue

«CodeList»

GenericLogicalSpaceFunctionValue

«CodeList»

GenericOccupiedSpaceFunctionValue

«CodeList» «CodeList»

GenericUnoccupiedSpaceFunctionValue GenericThematicSurfaceFunctionValue

«CodeList»

GenericLogicalSpaceUsageValue

«CodeList»

GenericOccupiedSpaceUsageValue

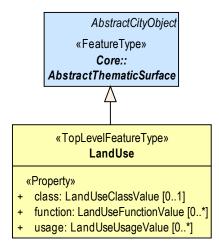
«CodeList»

GenericUnoccupiedSpaceUsageValue

«CodeList»

GenericThematicSurfaceUsageValue

## LandUse module



## LandUse module - Code lists

«CodeList»

LandUseClassValue

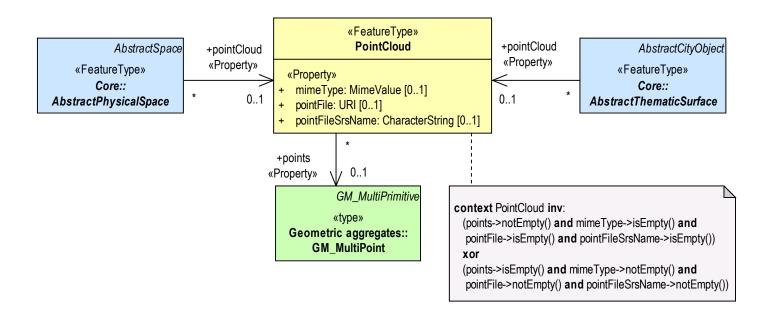
«CodeList»

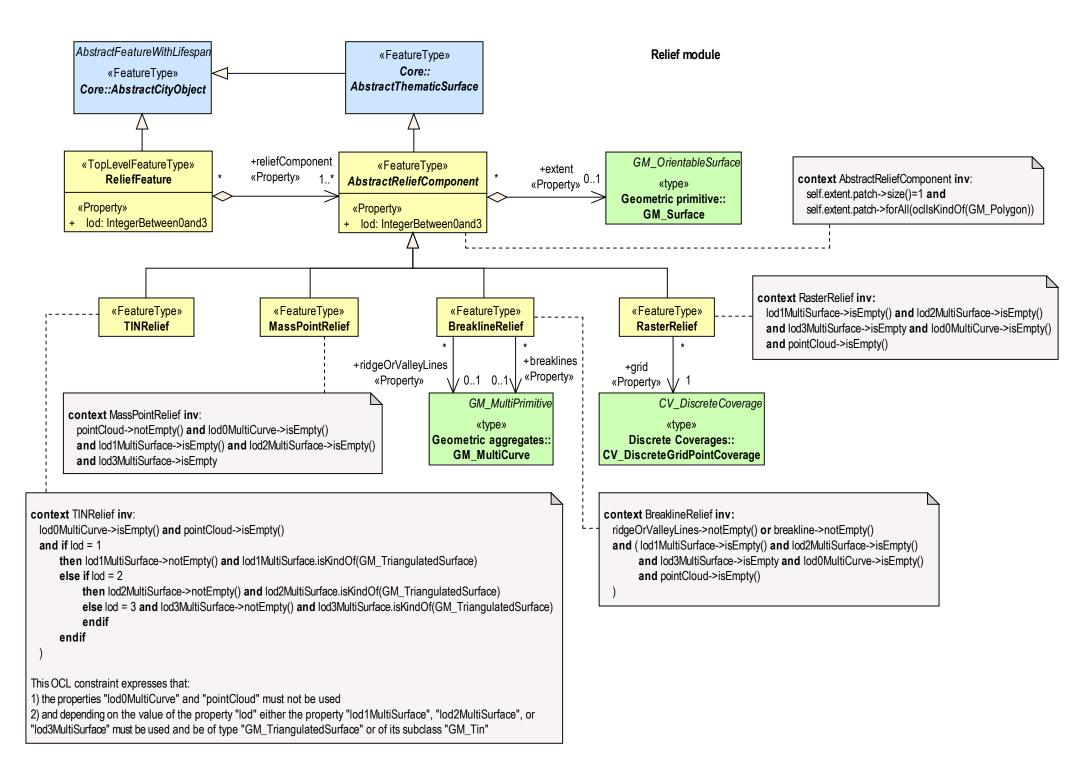
LandUseFunctionValue

«CodeList»

LandUseUsageValue

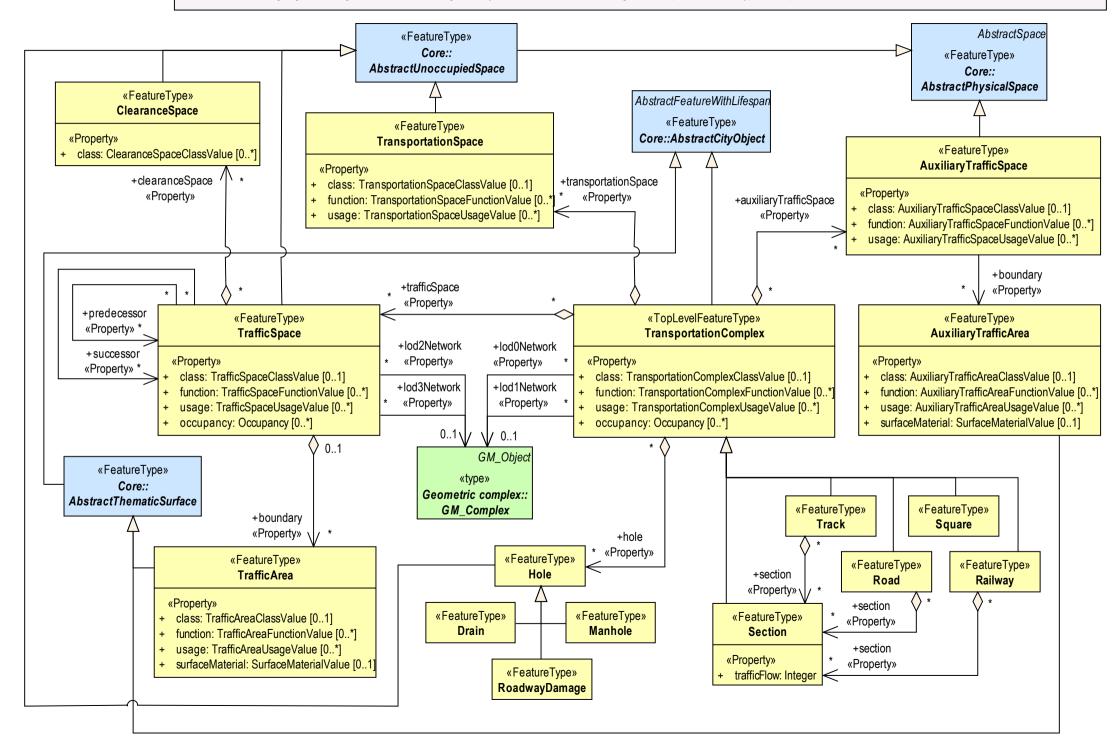
#### PointCloud module





Transportation module

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

«CodeList»

**TransportationComplexClassValue** 

«CodeList»

**TransportationComplexFunctionValue** 

«CodeList»

**TransportationComplexUsageValue** 

«CodeList» **TransportationSpaceClassValue** 

«CodeList»

**TransportationSpaceFunctionValue** 

«CodeList»

**TransportationSpaceUsageValue** 

«CodeList»

**TrafficSpaceClassValue** 

«CodeList»

**TrafficSpaceFunctionValue** 

«CodeList»

**TrafficSpaceUsageValue** 

«CodeList»

**TrafficAreaClassValue** 

«CodeList»

**TrafficAreaFunctionValue** 

«CodeList»

**TrafficAreaUsageValue** 

«CodeList»

**AuxiliaryTrafficSpaceClassValue** 

«CodeList»

**AuxiliaryTrafficSpaceFunctionValue** 

«CodeList»

**AuxiliaryTrafficSpaceUsageValue** 

«CodeList»

**AuxiliaryTrafficAreaClassValue** 

«CodeList»

**AuxiliaryTrafficAreaFunctionValue** 

«CodeList»

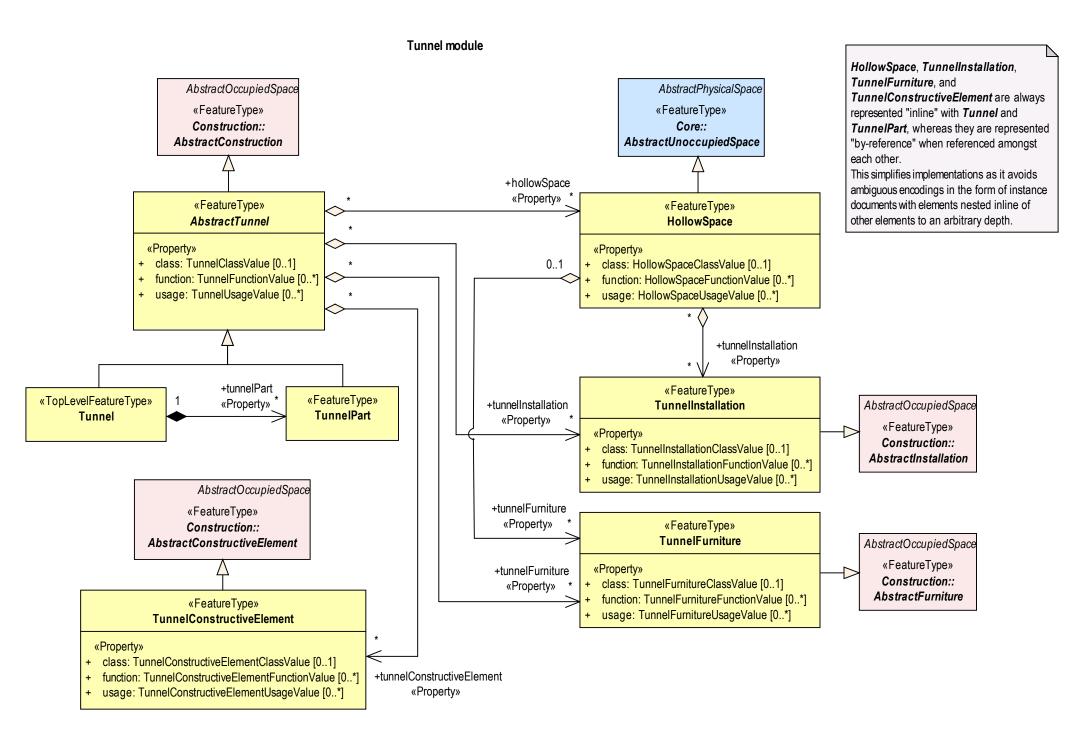
**AuxiliaryTrafficAreaUsageValue** 

«CodeList»

ClearanceSpaceClassValue

«CodeList»

SurfaceMaterialValue



## 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

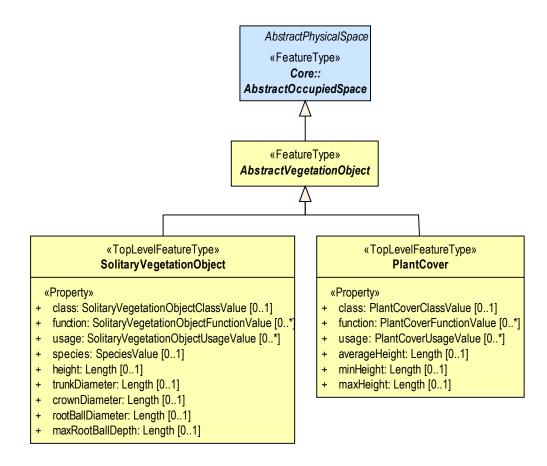
«CodeList»

HollowSpaceUsageValue

«CodeList»

TunnelConstructiveElementUsageValue

## Vegetation module



## Vegetation module - Code lists

«CodeList»

SolitaryVegetationObjectClassValue

«CodeList»

PlantCoverClassValue

«CodeList»

Solitary Vegetation Object Function Value

«CodeList»

PlantCoverFunctionValue

«CodeList»

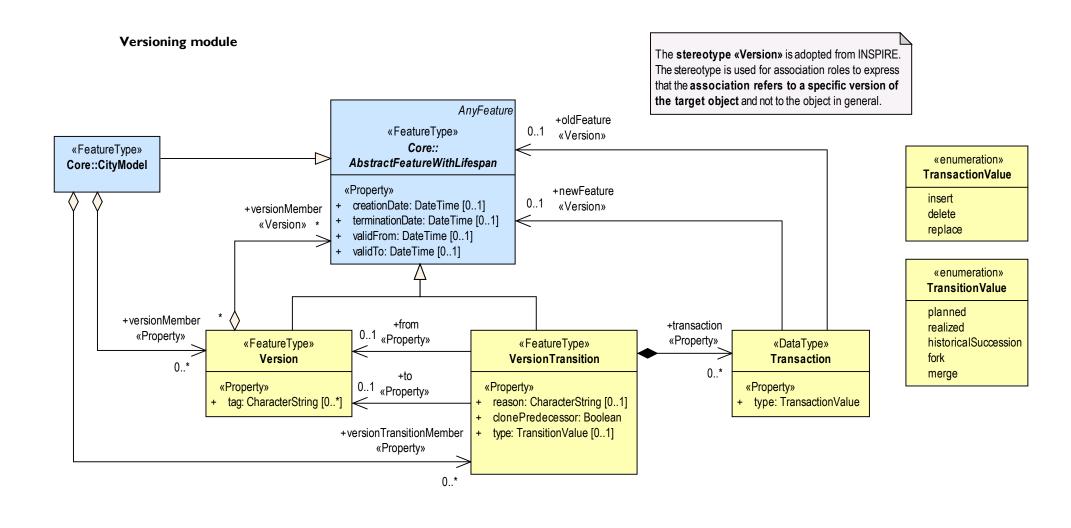
Solitary Vegetation Object Usage Value

«CodeList»

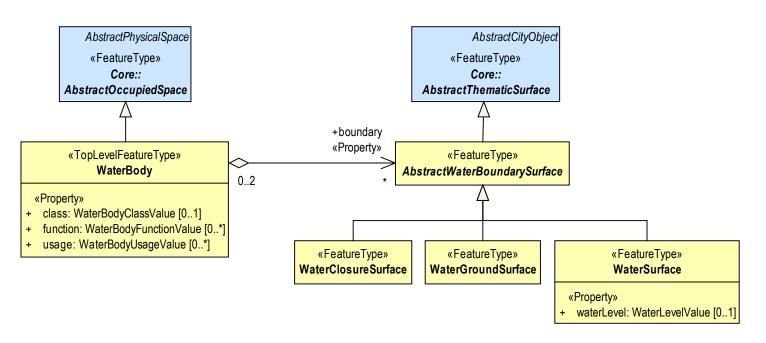
PlantCoverUsageValue

«CodeList»

SpeciesValue



## WaterBody module



# WaterBody module - Code lists

«CodeList»

WaterBodyClassValue

«CodeList»
WaterLevelValue

«CodeList»

WaterBodyFunctionValue

«CodeList»

WaterBodyUsageValue