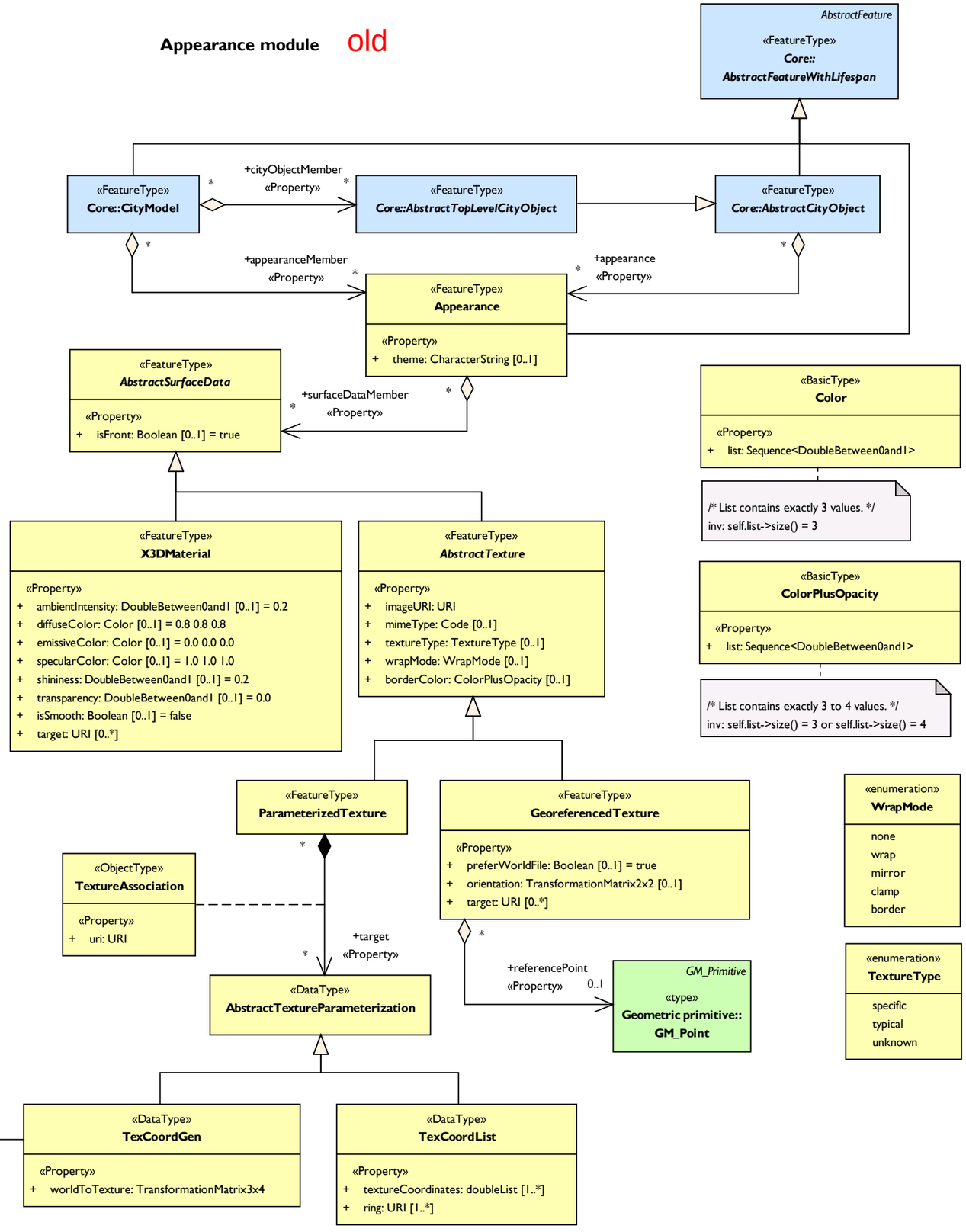
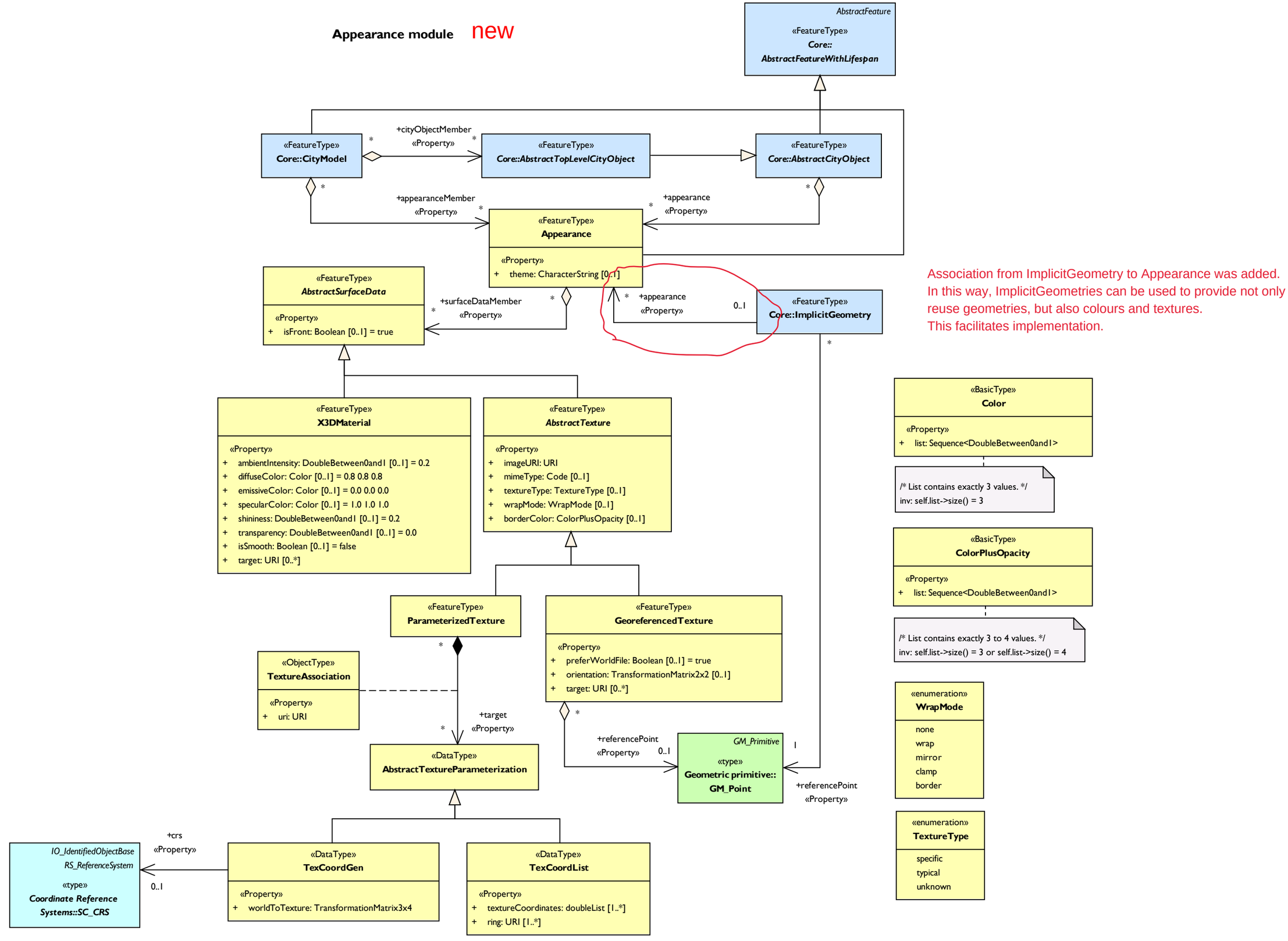
Appearance module old



These slides document the modifications applied to the CityGML 3.0 consolidated draft from 2018-03-21.

Appearance module new



Association from ImplicitGeometry to Appearance was added.
In this way, ImplicitGeometries can be used to provide not only reuse geometries, but also colours and textures.
This facilitates implementation.

Construction module old

AbstractConstruction is a so-called "mixin" class, i. e. during the mapping to GML its properties and associations are just copied to its subclass ***OtherConstruction***. This mechanism allows the handling of multiple inheritance, which is not supported by some platforms like XML. Hence, ***OtherConstruction*** is mapped to GML in a way that ***AbstractToplevelCityObject*** is its real superclass.

«FeatureType»
AbstractConstruction

«Property»
+ conditionOfConstruction: ConditionOfConstructionValue [0..1]
+ dateOfConstruction: Date [0..1]
+ dateOfRenovation: Date [0..*]
+ dateOfDemolition: Date [0..1]
+ elevation: Elevation [0..*]
+ heightAboveGround: HeightAboveGround [0..*]

AbstractCityObject
«FeatureType»
Core::
AbstractTopLevelCityObject

«FeatureType»
OtherConstruction

«Property»
+ class: Code
+ function: Code [0..*]
+ usage: Code [0..*]

AbstractPhysicalSpace
«FeatureType»
Core::
AbstractOccupiedSpace

«FeatureType»
AbstractConstructiveElement

«FeatureType»
AbstractFurniture

«FeatureType»
AbstractInstallation

«Property»
+ relationToConstruction: RelationToConstruction [0..1]

+constructionSpace
«Property»
1

«FeatureType»
ConstructionSpace

+bounds
«Property»

0..2
{subsets
bounds}

AbstractUnoccupiedSpace
«FeatureType»
Core::AbstractVoid

+opening
«Property»
*

AbstractSpaceBoundary
«FeatureType»
Core::
AbstractThematicSurface

+opening
«Property»
*

AbstractSpaceBoundary
«FeatureType»
Core::
AbstractVoidSurface

+boundary
«Property»
*
{subsets
boundary}

+bounds
«Property»
*
{subsets
bounds}

+boundary
«Property»
*
{subsets
boundary}

«FeatureType»
AbstractConstructionSurface

«FeatureType»
AbstractConstructionOpeningSurface

«FeatureType»
AbstractConstructionVoid

«FeatureType»
GroundSurface

«FeatureType»
RoofSurface

«FeatureType»
FloorSurface

«FeatureType»
OuterFloorSurface

«FeatureType»
CeilingSurface

«FeatureType»
OuterCeilingSurface

«FeatureType»
WallSurface

«FeatureType»
InteriorWallSurface

«FeatureType»
DoorSurface

«FeatureType»
WindowSurface

«FeatureType»
Door

«FeatureType»
Window

*
+address
«Property»
*

«FeatureType»
Core::Address

*
+address
«Property»
*

It needs to be discussed further within the SWG, whether a ***Door*** is a ***Void*** or an ***Opening***.

«CodeList»
ConditionOfConstructionValue

+ declined
+ demolished
+ functional
+ projected
+ ruin
+ underConstruction

«enumeration»
RelationToConstruction

inside
outside
bothInsideAndOutside

«DataType»
Elevation

«Property»
+ elevationReference: ElevationReferenceValue
+ elevationValue: DirectPosition

«DataType»
HeightAboveGround

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

«CodeList»
ElevationReferenceValue

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

«CodeList»
HeightStatusValue

+ estimated
+ measured

Construction module new

AbstractConstruction is a so-called "mixin" class, i. e. during the mapping to GML its properties and associations are just copied to its subclass ***OtherConstruction***. This mechanism allows the handling of multiple inheritance, which is not supported by some platforms like XML. Hence, ***OtherConstruction*** is mapped to GML in a way that ***AbstractToplevelCityObject*** is its real superclass.

The code lists were changed into enumerations as we do not want them to be extensible.

The composition was modified, it relates now AbstractConstructiveElement and AbstractVoid.

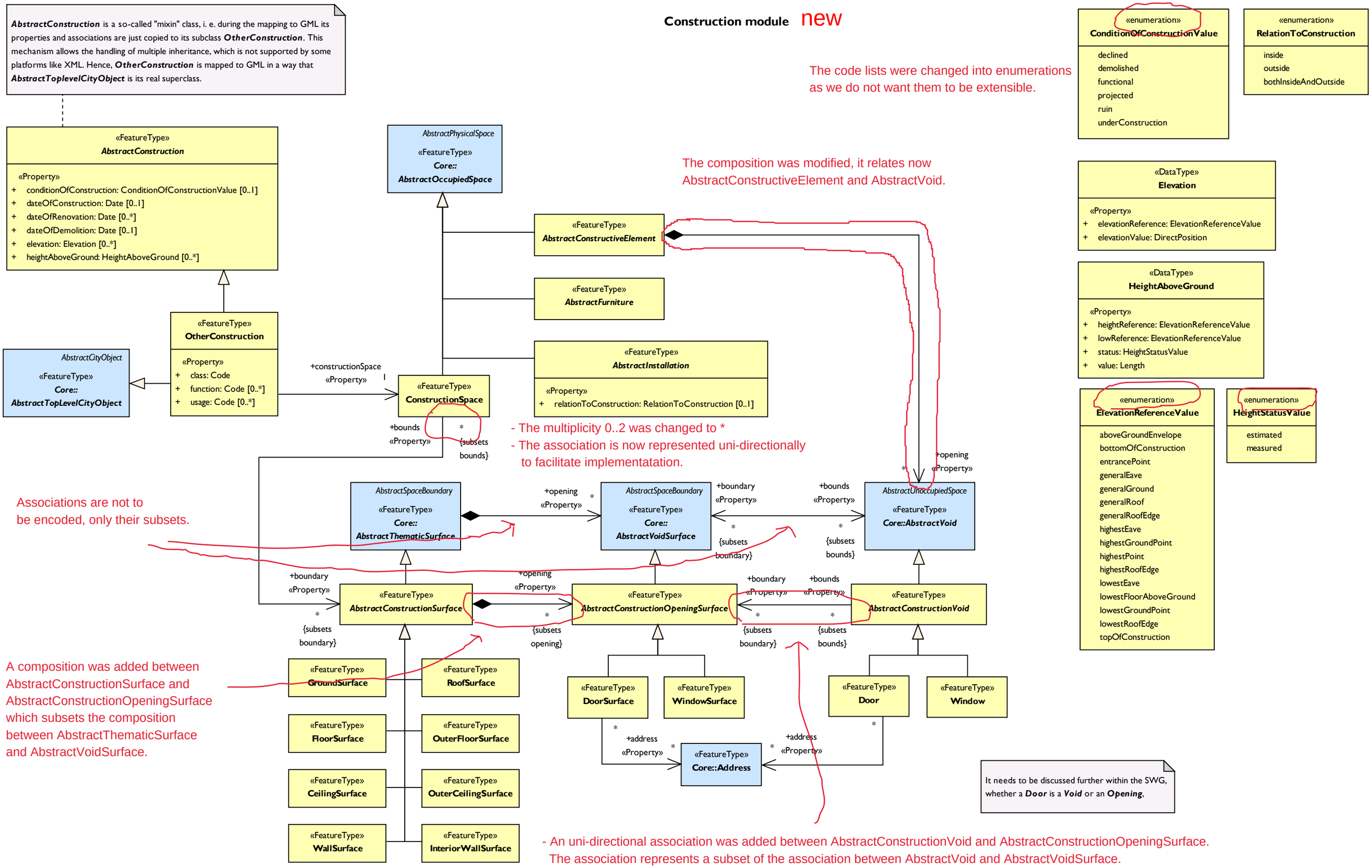
- The multiplicity 0..2 was changed to *
- The association is now represented uni-directionally to facilitate implementatation.

Associations are not to be encoded, only their subsets.

A composition was added between AbstractConstructionSurface and AbstractConstructionOpeningSurface which subsets the composition between AbstractThematicSurface and AbstractVoidSurface.

- An uni-directional association was added between AbstractConstructionVoid and AbstractConstructionOpeningSurface. The association represents a subset of the association between AbstractVoid and AbstractVoidSurface.
- The association is byReference only to facilitate implementation.

It needs to be discussed further within the SWG, whether a ***Door*** is a ***Void*** or an ***Opening***,



Building module old

AbstractBuilding is a so-called "mixin" class, i. e. during the mapping to GML its properties and associations are just copied to its subclasses (***Building*** and ***BuildingPart***). This mechanism allows the handling of multiple inheritance, which is not supported by some platforms like XML. Hence, ***Building*** is mapped to GML in a way that ***AbstractToplevelCityObject*** is its real superclass and ***AbstractCityObject*** is the real superclass of ***BuildingPart.***

AbstractOccupiedSpace
«FeatureType»
Construction::
ConstructionSpace

«FeatureType»
Construction::
AbstractConstruction

AbstractSpace
«FeatureType»
Core::
AbstractLogicalSpace

+buildingSpace
«Property»
1..*

+buildingSubdivision
«Property»
*

«FeatureType»
AbstractBuilding
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ roofType: Code [0..1]
+ storeysAboveGround: Integer [0..1]
+ storeysBelowGround: Integer [0..1]
+ storeyHeightsAboveGround: MeasureOrNilReasonList [0..1]
+ storeyHeightsBelowGround: MeasureOrNilReasonList [0..1]

«FeatureType»
AbstractBuildingSubdivision
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ elevation: HeightAboveGround [0..*]
+ sortKey: Real [0..1]

+address
«Property»
«FeatureType»
Core::Address
+address
«Property»

AbstractPhysicalSpace
«FeatureType»
Core::
AbstractUnoccupiedSpace

«FeatureType»
BuildingUnit
«FeatureType»
Storey
+buildingUnit
«Property»
+storey
«Property»

+interiorRoom
«Property»

«FeatureType»
Room
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ elevation: HeightAboveGround [0..*]

+room
«Property»

«FeatureType»
Core::
AbstractTopLevelCityObject

AbstractFeatureWithLifespan
«FeatureType»
Core::AbstractCityObject

«FeatureType»
Building
+buildingPart
«Property»
«FeatureType»
BuildingPart

+roomInstallation
«Property»

+buildingSubdivisionInstallation
«Property»

«FeatureType»
BuildingInstallation
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]

+buildingInstallation
«Property»

AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractInstallation

«FeatureType»
BuildingConstructiveElement
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]

AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractConstructiveElement

+buildingConstructiveElement
«Property»

+buildingSubdivisionConstructiveElement
«Property»

+interiorFurniture
«Property»

«FeatureType»
BuildingFurniture
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]

AbstractOccupiedSpace
«FeatureType»
Construction::
AbstractFurniture

+buildingFurniture
«Property»

+buildingSubdivisionFurniture
«Property»

Building module new

An association from Storey to AbstractConstructionSurface was added.

AbstractBuilding is a so-called "mixin" class, i. e. during the mapping to GML its properties and associations are just copied to its subclasses (***Building*** and ***BuildingPart***). This mechanism allows the handling of multiple inheritance, which is not supported by some platforms like XML. Hence, ***Building*** is mapped to GML in a way that ***AbstractToplevelCityObject*** is its real superclass and ***AbstractCityObject*** is the real superclass of ***BuildingPart.***

«FeatureType» ***Construction::AbstractConstruction***

AbstractOccupiedSpace «FeatureType» **Construction::ConstructionSpace**

+bounds «Property» * {subsets bounds}

+boundary «Property» * {subsets boundary}

AbstractThematicSurface «FeatureType» ***Construction::AbstractConstructionSurface***

+boundary «Property» * byReference

+buildingSpace «Property» 1..* inlineOrByReference

«FeatureType» ***AbstractBuilding***

«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ roofType: Code [0..1]
+ storeysAboveGround: Integer [0..1]
+ storeysBelowGround: Integer [0..1]
+ storeyHeightsAboveGround: MeasureOrNilReasonList [0..1]
+ storeyHeightsBelowGround: MeasureOrNilReasonList [0..1]

1 +buildingSubdivision «Property» * inline

AbstractSpace «FeatureType» ***Core::AbstractLogicalSpace***

+address «Property» * inlineOrByReference

«FeatureType» **Core::Address**

* inlineOrByReference +address «Property»

«FeatureType» ***AbstractBuildingSubdivision***

«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ elevation: Elevation [0..*]
+ sortKey: Real [0..1]

Room*, *BuildingInstallation*, *BuildingFurniture*, *AbstractBuildingSubdivision and ***BuildingConstructiveElement*** are always represented "inline" with ***Building*** and ***BuildingPart***, whereas they are represented "by-reference" when referenced amongst each other.

This is done in the same way also in the Bridge and Tunnel modules.

Inconsistency between attribute name "elevation" and attribute type "HeightAboveGround" -> should be room height for Rooms and elevation for Subdivisions

«FeatureType» ***Core::AbstractTopLevelCityObject***

AbstractFeatureWithLifespan «FeatureType» ***Core::AbstractCityObject***

AbstractPhysicalSpace «FeatureType» ***Core::AbstractUnoccupiedSpace***

+interiorRoom «Property» * inline

«FeatureType» **Room**

«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ roomHeight: RoomHeight [0..*]

«FeatureType» **BuildingUnit** * byReference +buildingUnit «Property»

«FeatureType» **Storey** * byReference +storey «Property»

+room «Property» * byReference

«FeatureType» **Building** * +buildingPart «Property» * inlineOrByReference «FeatureType» **BuildingPart**

«dataType» **RoomHeight**

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

«enumeration» **ElevationReferenceValue**

to be determined

HeightAboveGround cannot be used for Rooms, as the enumeration values to not fit. A similar data type + enumeration needs to be defined for Room.

«FeatureType» **BuildingConstructiveElement**

«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]

AbstractOccupiedSpace «FeatureType» ***Construction::AbstractConstructiveElement***

inline +buildingConstructiveElement «Property» * +buildingSubdivisionConstructiveElement «Property» byReference

+roomInstallation «Property» * byReference

+buildingSubdivisionInstallation «Property» * byReference

+buildingInstallation «Property» * inline

«FeatureType» **BuildingInstallation**

«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]

AbstractOccupiedSpace «FeatureType» ***Construction::AbstractInstallation***

+interiorFurniture «Property» * byReference

«FeatureType» **BuildingFurniture**

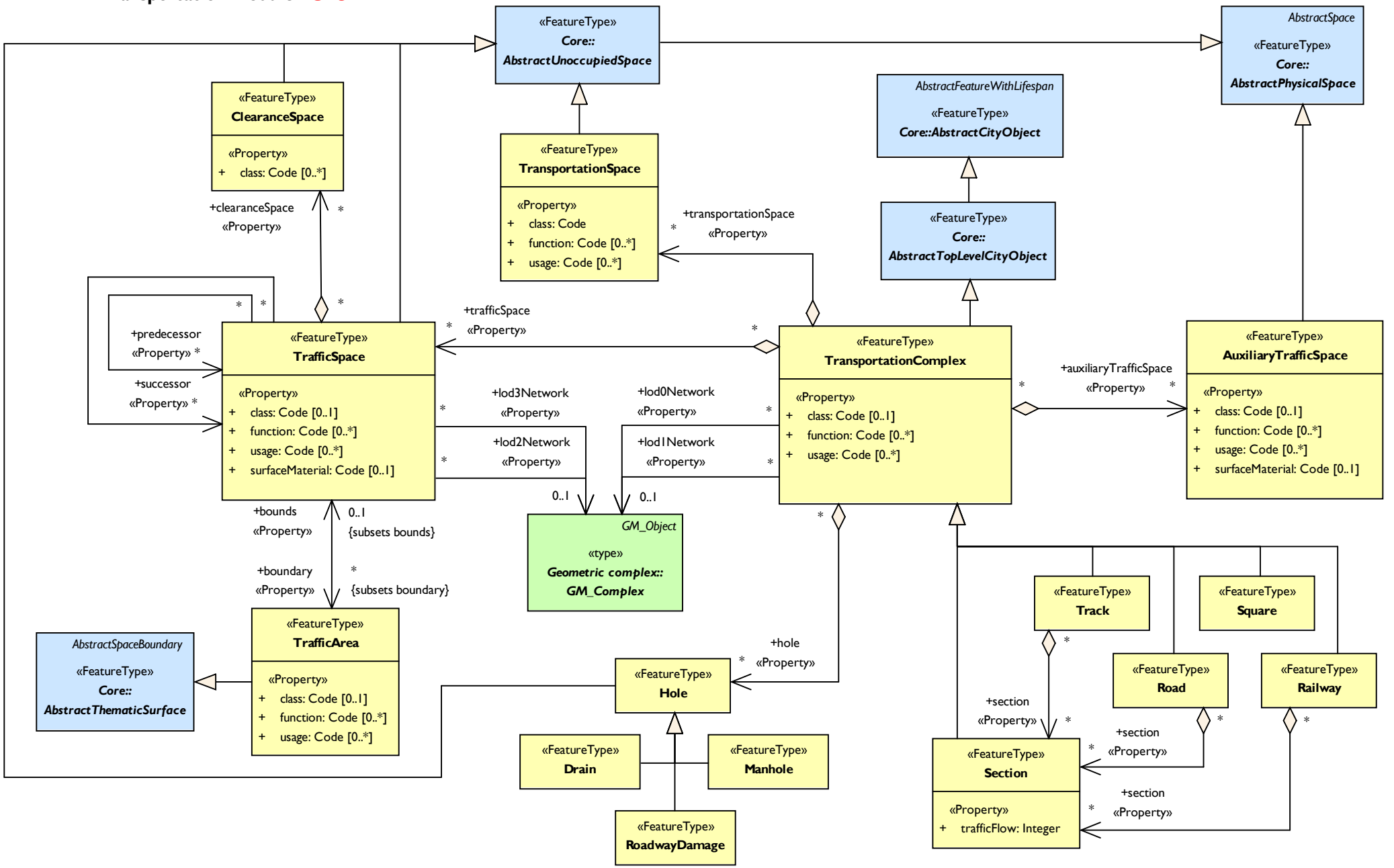
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]

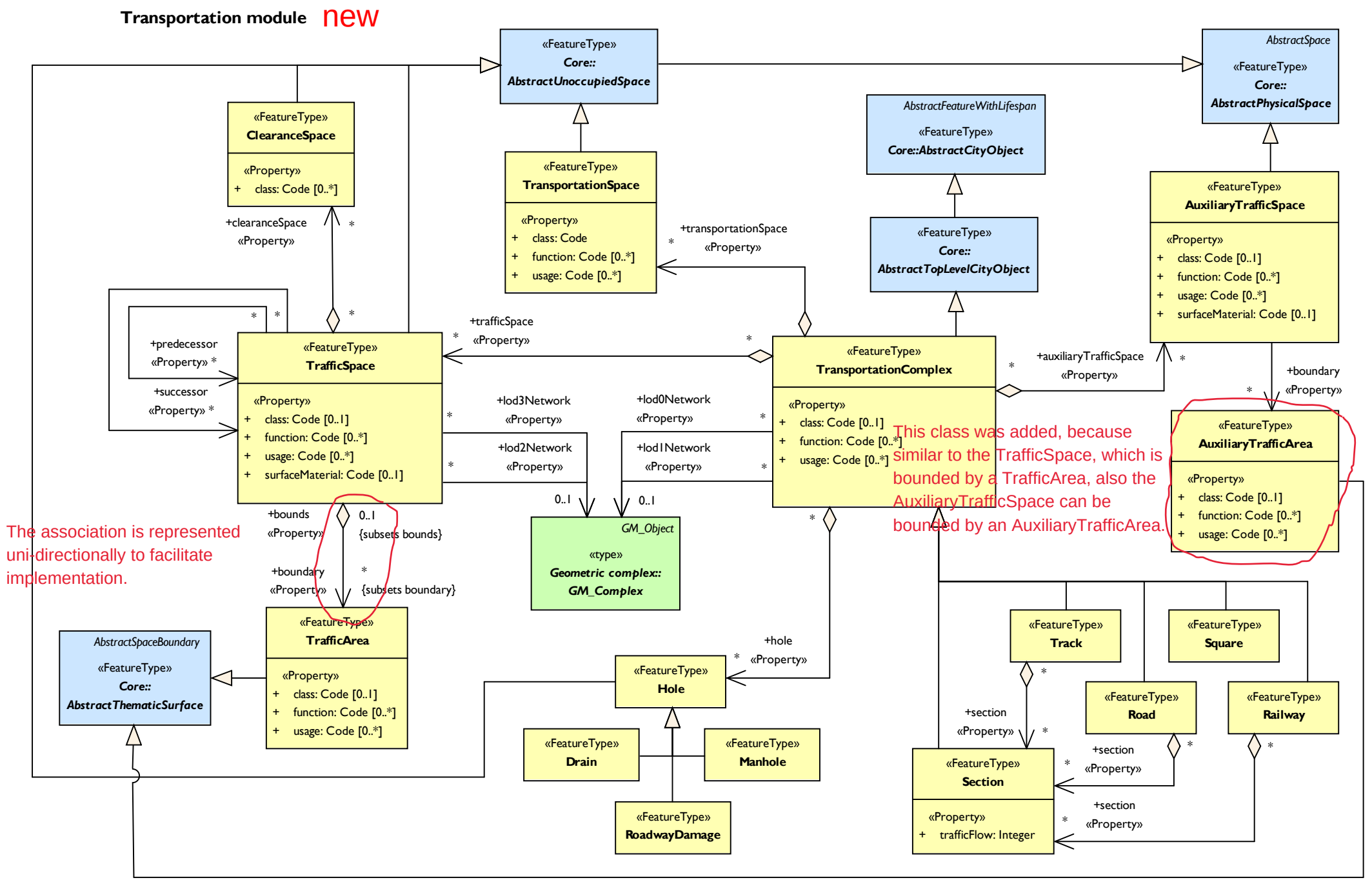
AbstractOccupiedSpace «FeatureType» ***Construction::AbstractFurniture***

+buildingFurniture «Property» * inline

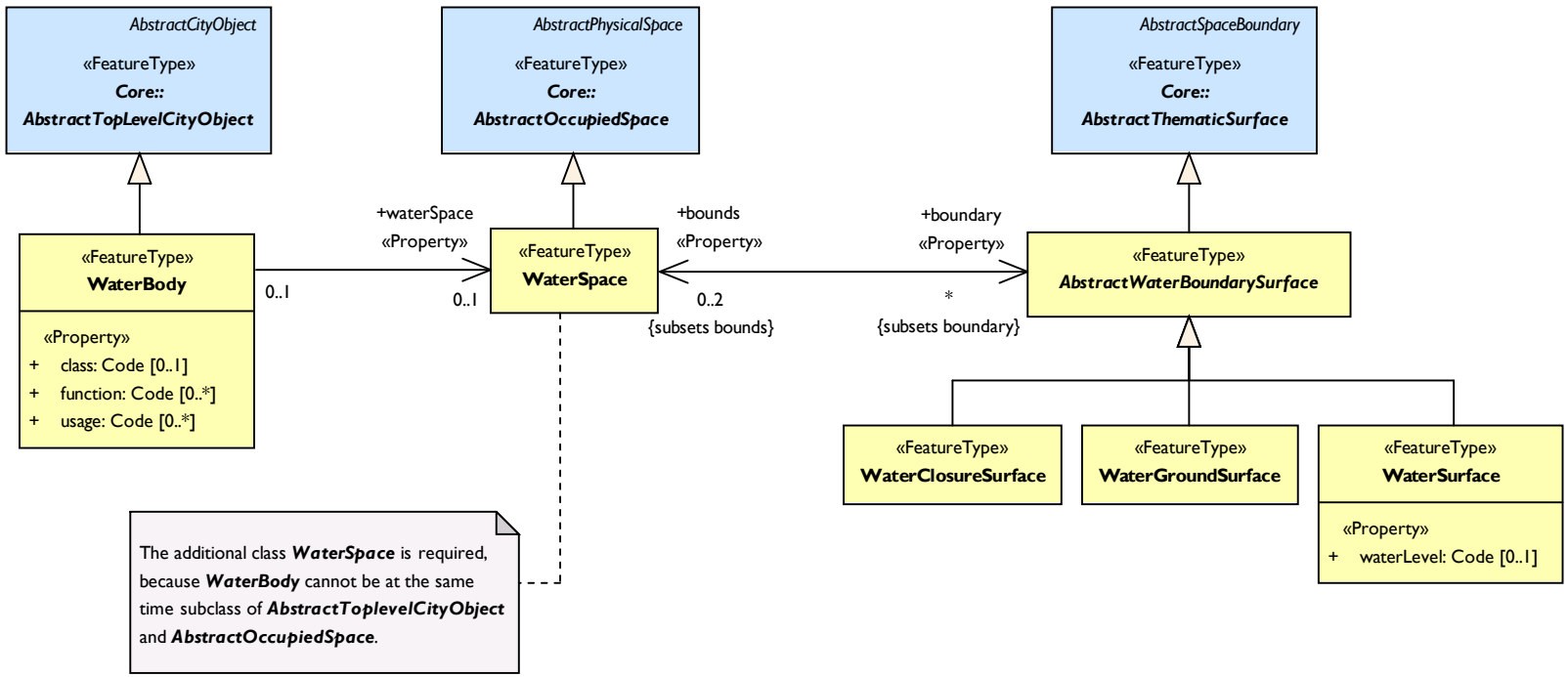
+buildingSubdivisionFurniture «Property» * byReference

Transportation module old

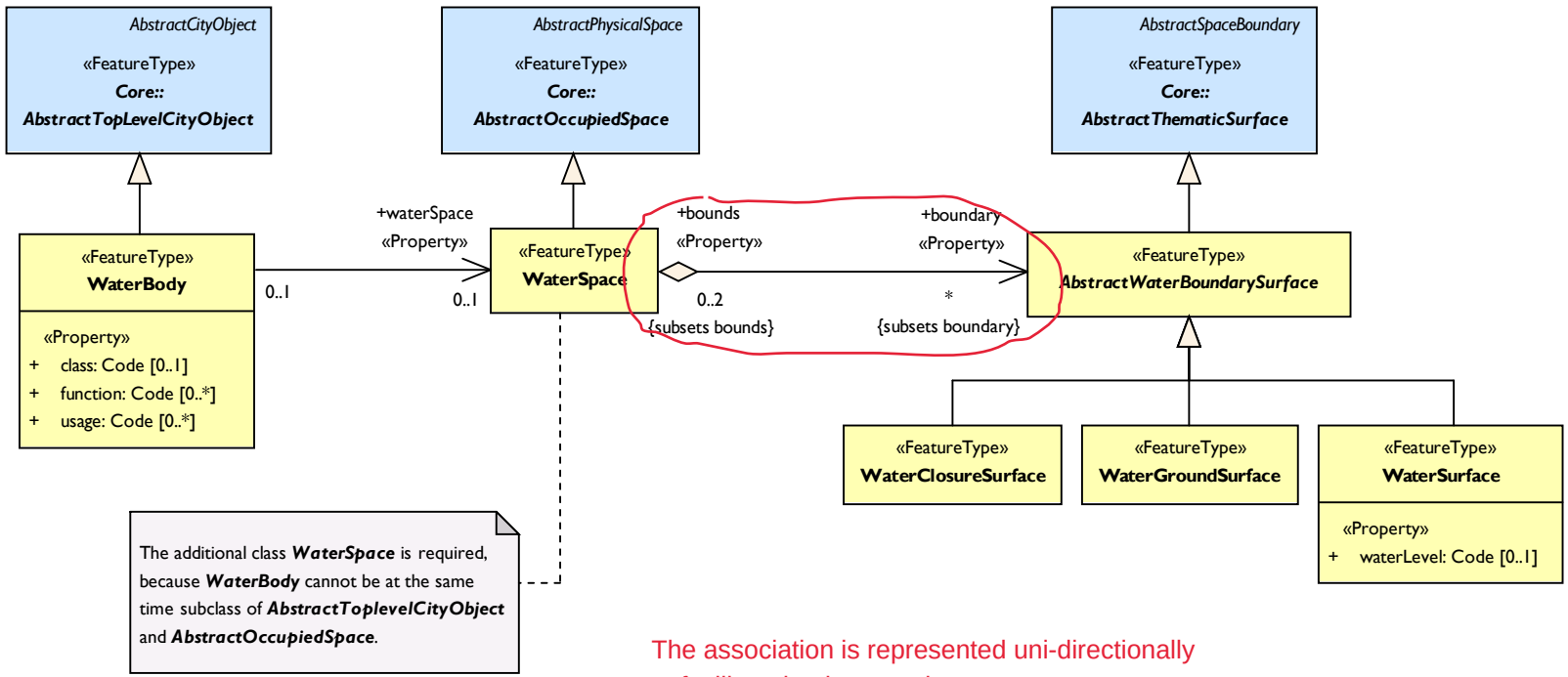



Transportation module
new
«FeatureType»
Core::
AbstractUnoccupiedSpace
AbstractSpace
«FeatureType»
Core::
AbstractPhysicalSpace
«FeatureType»
ClearanceSpace
«Property»
+ class: Code [0..*]
+clearanceSpace
«Property»
«FeatureType»
TransportationSpace
«Property»
+ class: Code
+ function: Code [0..*]
+ usage: Code [0..*]
+transportationSpace
«Property»
AbstractFeatureWithLifespan
«FeatureType»
Core::AbstractCityObject
«FeatureType»
Core::
AbstractTopLevelCityObject
«FeatureType»
AuxiliaryTrafficSpace
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ surfaceMaterial: Code [0..1]
+predecessor
«Property»
+successor
«Property»
«FeatureType»
TrafficSpace
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+ surfaceMaterial: Code [0..1]
+trafficSpace
«Property»
«FeatureType»
TransportationComplex
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+auxiliaryTrafficSpace
«Property»
+boundary
«Property»
«FeatureType»
AuxiliaryTrafficArea
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
+lod3Network
«Property»
+lod2Network
«Property»
+lod0Network
«Property»
+lod1Network
«Property»
0..1
0..1
GM_Object
«type»
Geometric complex::
GM_Complex
+bounds
«Property»
0..1
{subsets bounds}
+boundary
«Property»
*
{subsets boundary}
«FeatureType»
TrafficArea
«Property»
+ class: Code [0..1]
+ function: Code [0..*]
+ usage: Code [0..*]
AbstractSpaceBoundary
«FeatureType»
Core::
AbstractThematicSurface
+hole
«Property»
«FeatureType»
Hole
«FeatureType»
Drain
«FeatureType»
Manhole
«FeatureType»
RoadwayDamage
«FeatureType»
Track
«FeatureType»
Square
«FeatureType»
Road
«FeatureType»
Railway
+section
«Property»
«FeatureType»
Section
«Property»
+ trafficFlow: Integer
+section
«Property»
+section
«Property»
The association is represented uni-directionally to facilitate implementation.
This class was added, because similar to the TrafficSpace, which is bounded by a TrafficArea, also the AuxiliaryTrafficSpace can be bounded by an AuxiliaryTrafficArea.

WaterBody module old



WaterBody module new



The association is represented uni-directionally to facilitate implementation.