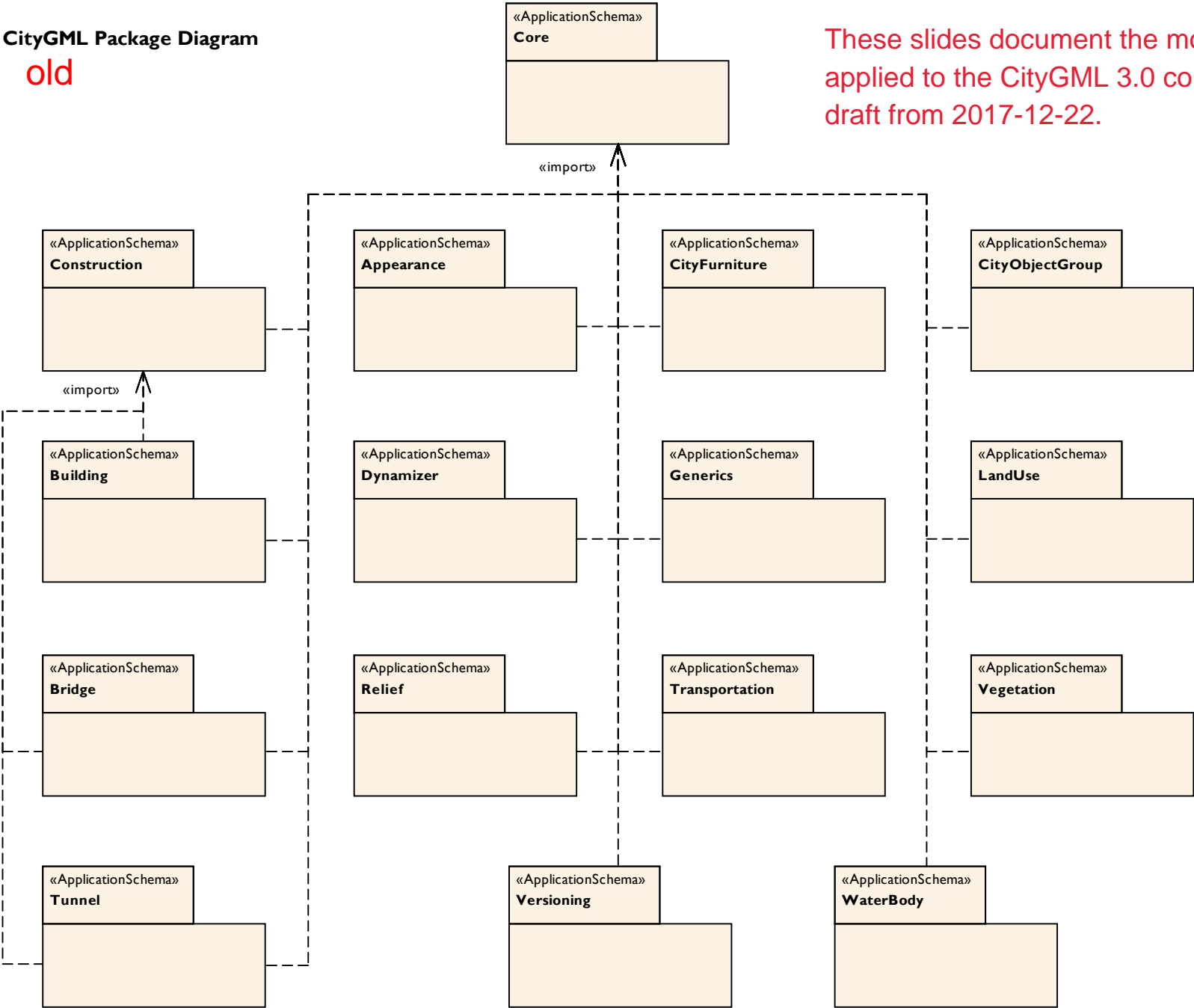


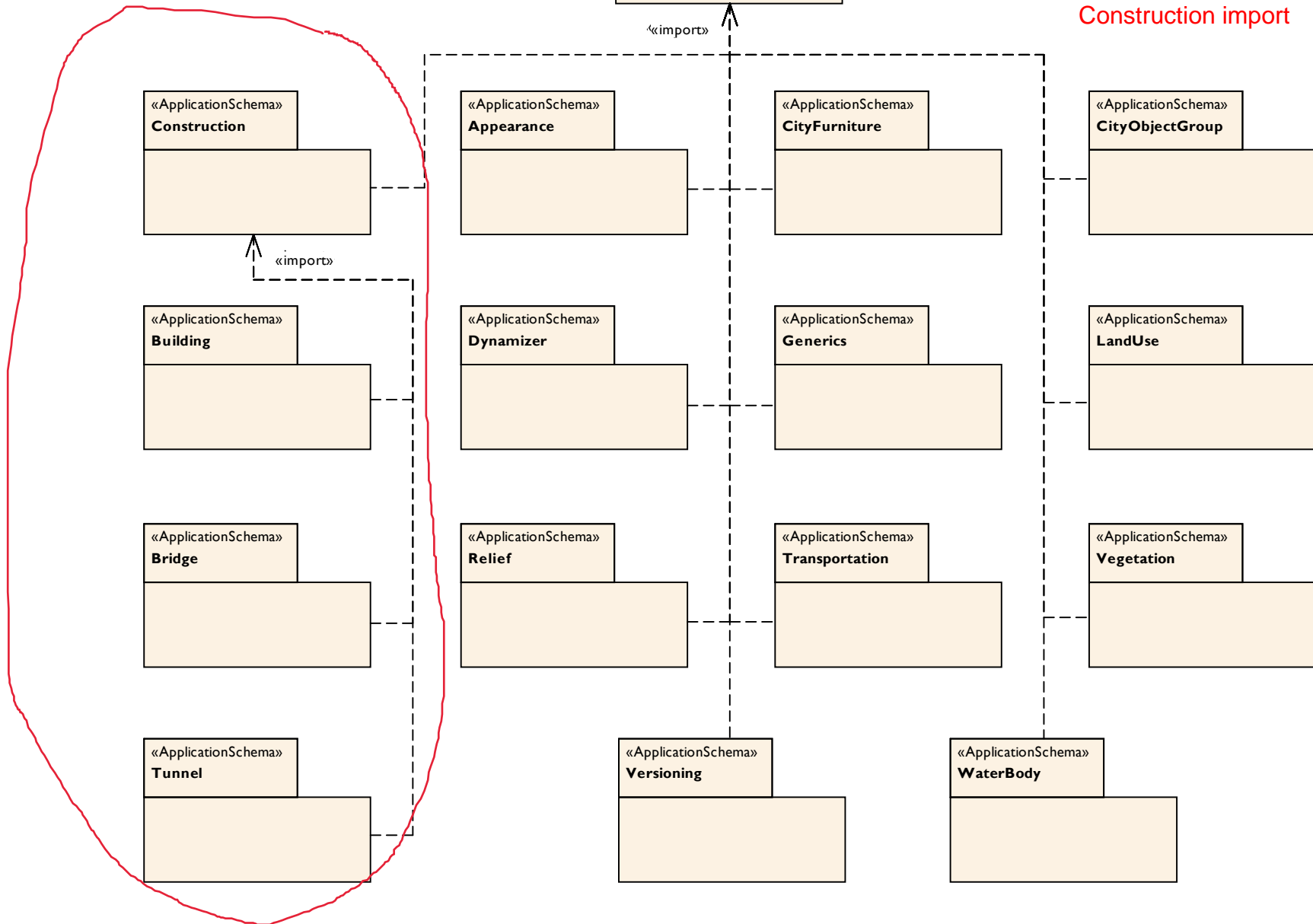
CityGML Package Diagram
old

These slides document the modifications
applied to the CityGML 3.0 consolidated
draft from 2017-12-22.



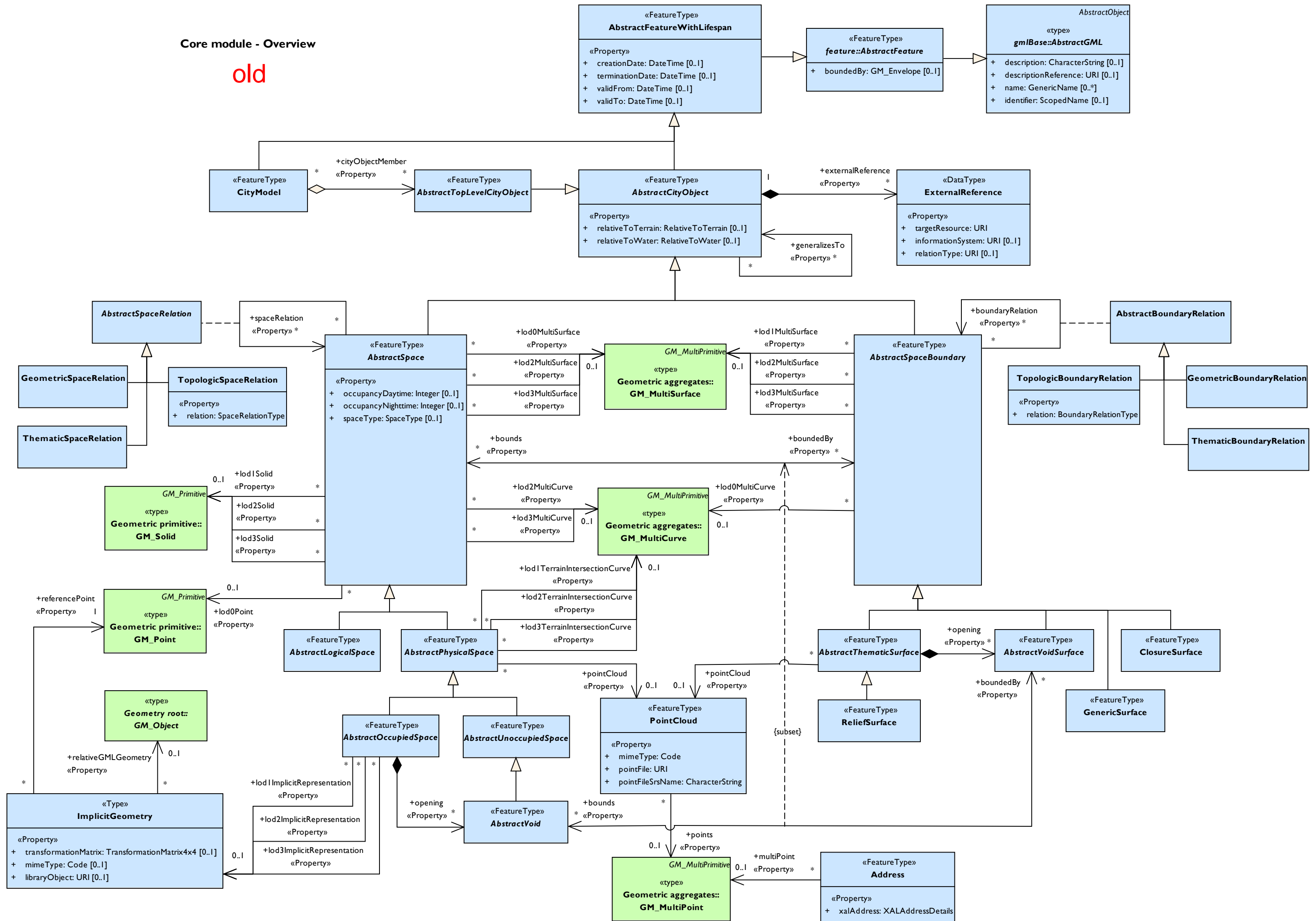
CityGML Package Diagram

new



The import relationship is transitive
-> Building, Tunnel and Bridge do not
need to import Core, as Core is already
imported transitively through the
Construction import

old

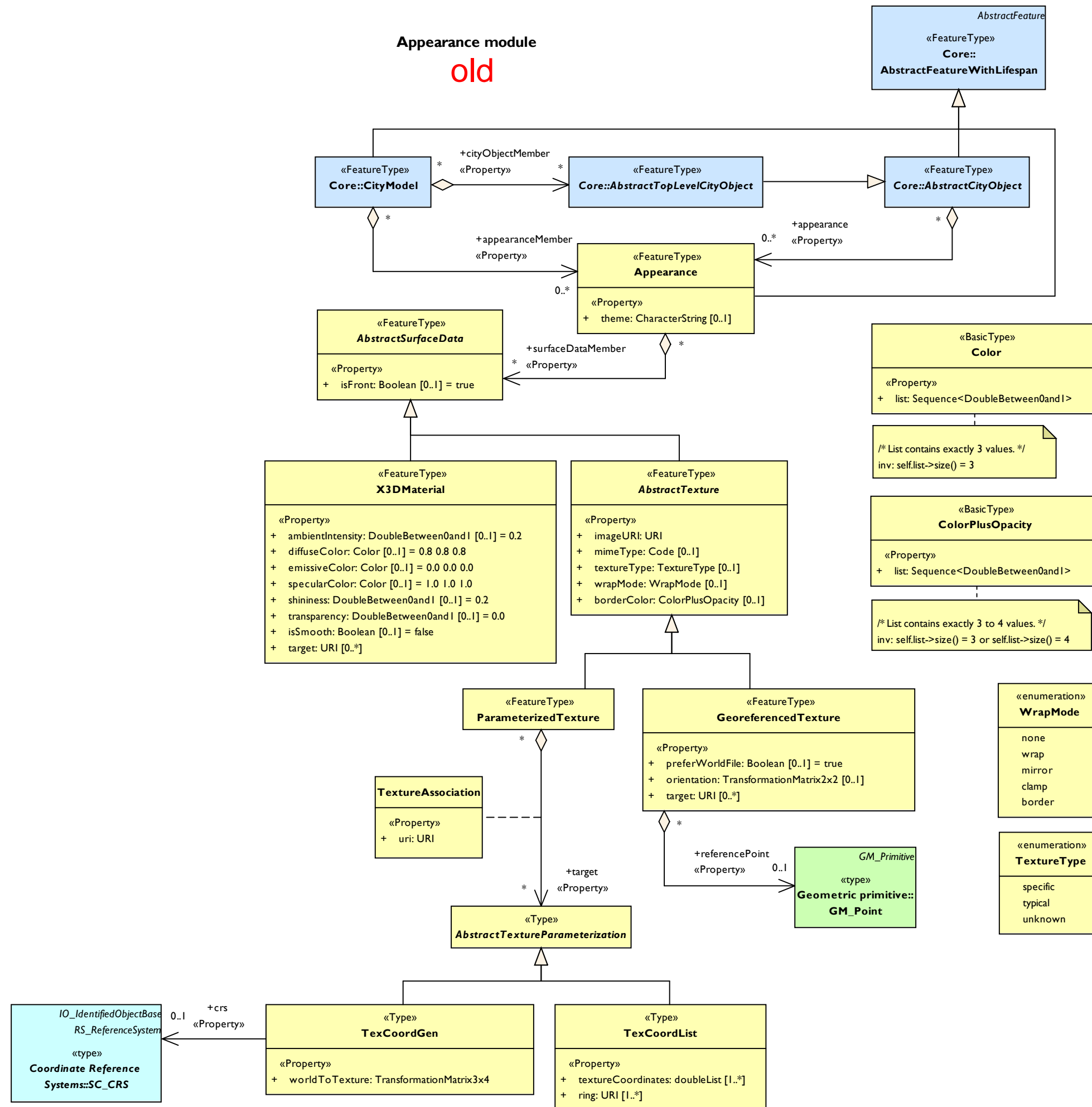


new



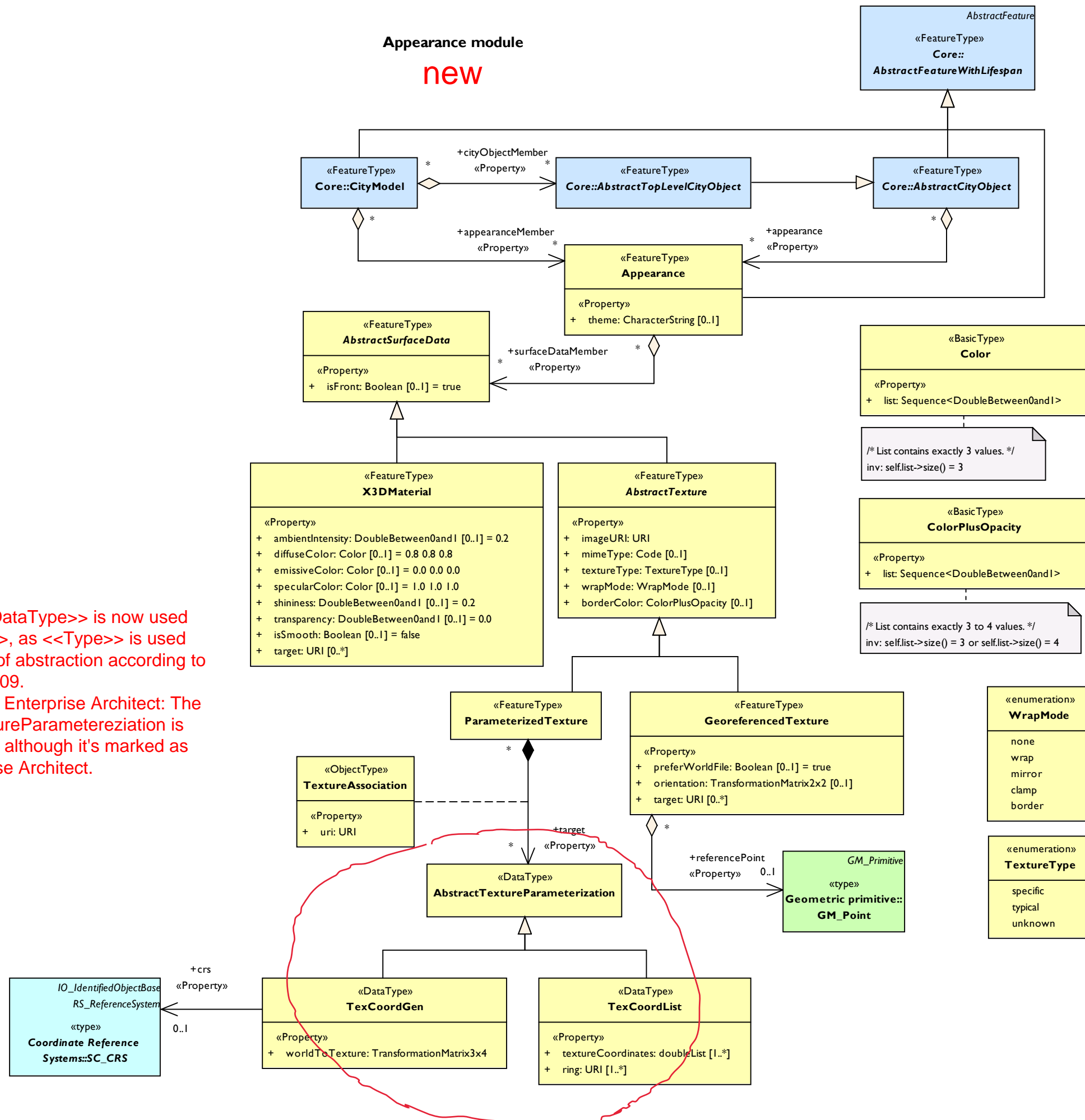
Appearance module

old



The stereotype <<DataType>> is now used instead of <<Type>>, as <<Type>> is used on a different level of abstraction according to ISO 19103 and 19109.

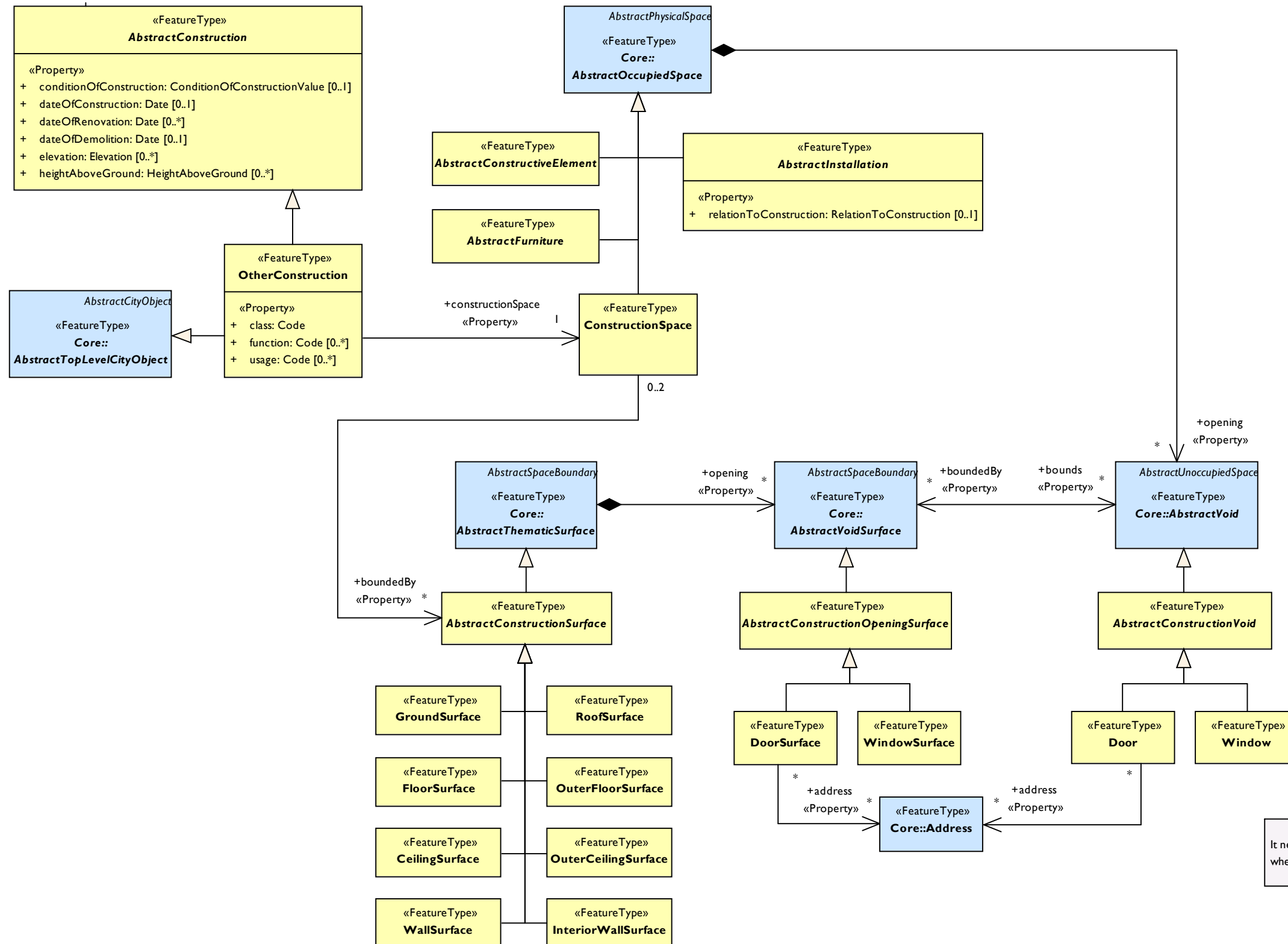
-> There is a bug in Enterprise Architect: The name AbstractTextureParameterization is not shown in italics, although it's marked as abstract in Enterprise Architect.



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.

Construction module

old



«CodeList» ConditionOfConstructionValue
+ declined
+ demolished
+ functional
+ projected
+ ruin
+ underConstruction

«enumeration» RelationToConstruction
inside
outside
bothInsideAndOutside

«DataType» Elevation
«Property»
+ elevationReference: ElevationReferenceValue
+ elevationValue: DirectPosition

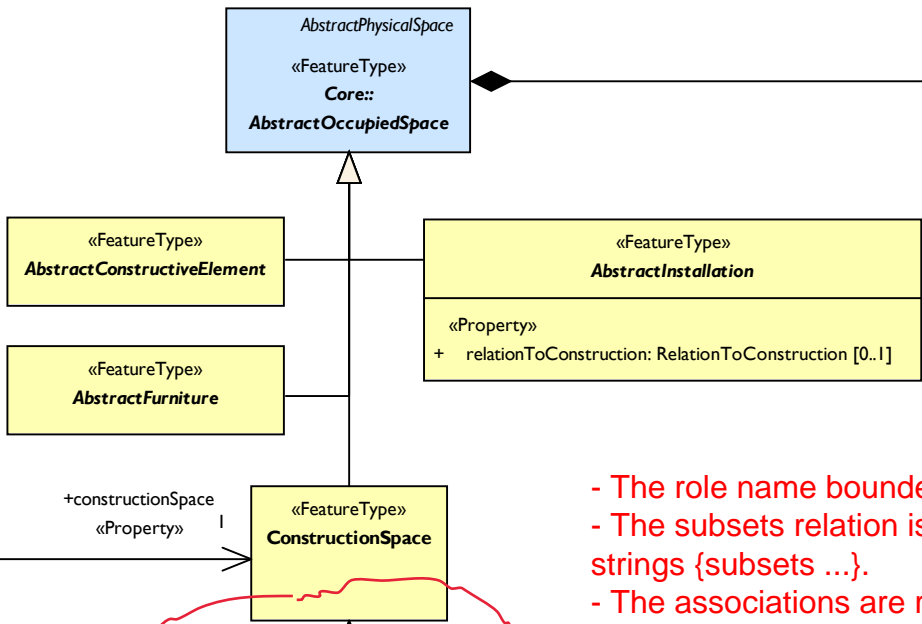
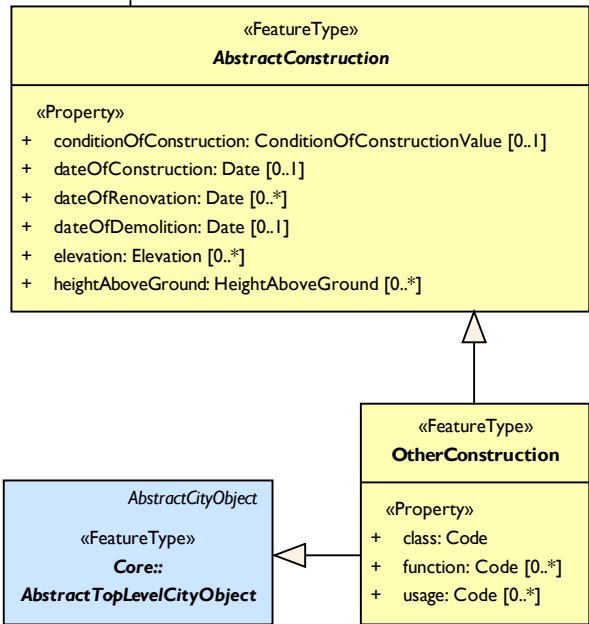
«DataType» HeightAboveGround
«Property»
+ heightReference: ElevationReferenceValue
+ lowReference: ElevationReferenceValue
+ status: HeightStatusValue
+ value: Length

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

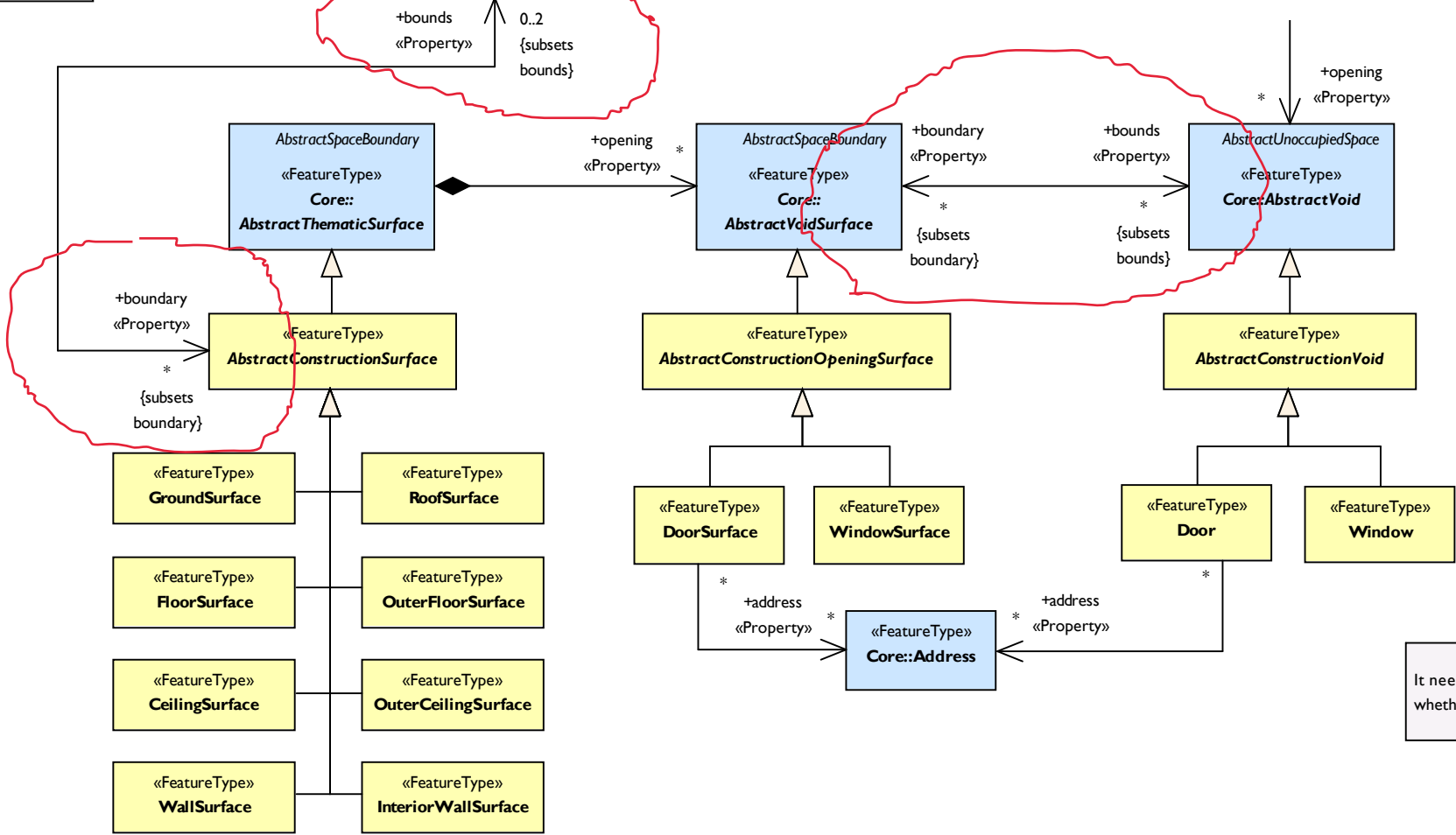
«CodeList» HeightStatusValue
+ estimated
+ measured

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

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 role name **boundedBy** is replaced by **boundary**.
- The subsets relation is represented through property strings {subsets ...}.
- The associations are represented bidirectionally.



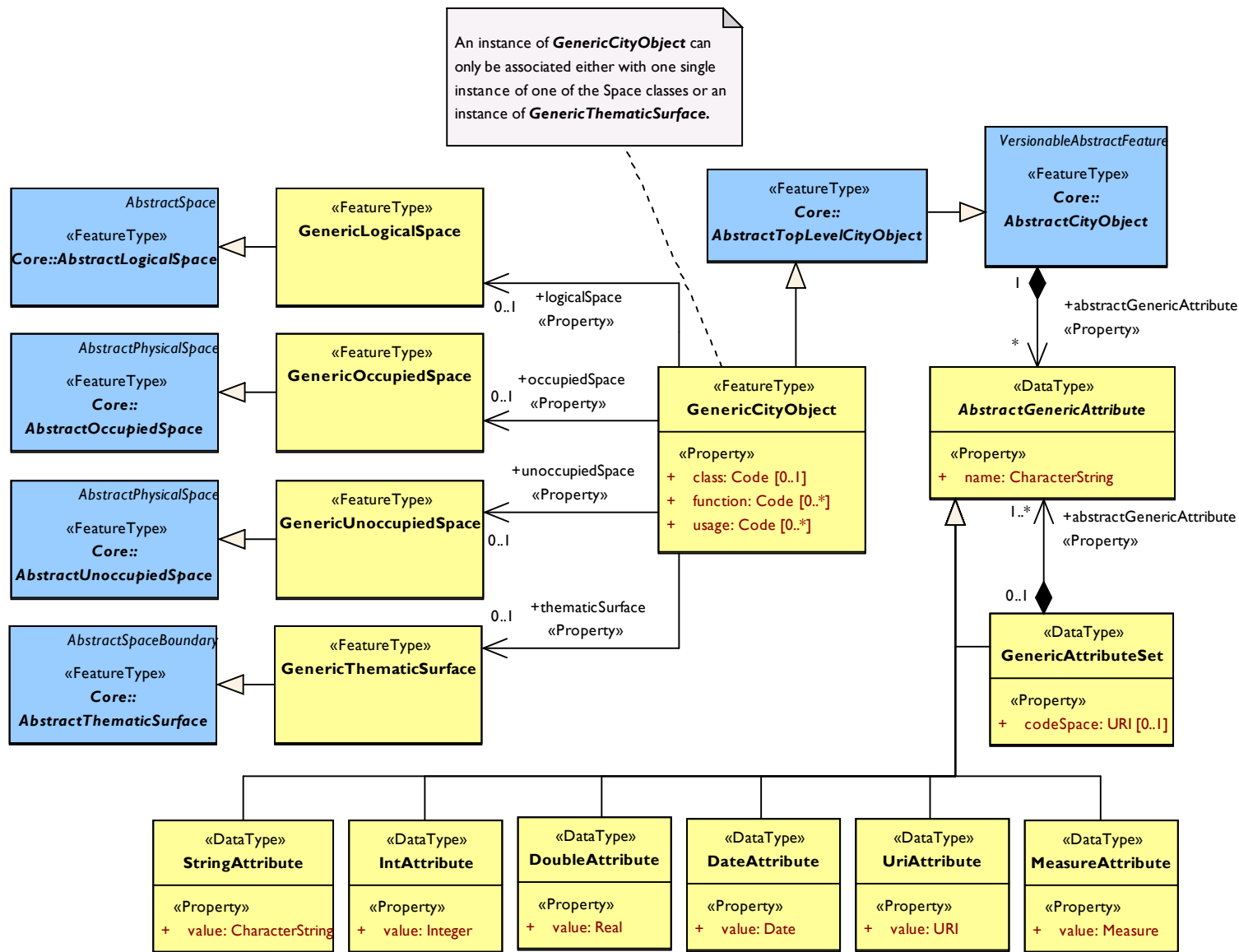
«CodeList» ConditionOfConstructionValue	«enumeration» RelationToConstruction
+ declined + demolished + functional + projected + ruin + underConstruction	inside outside bothInsideAndOutside

«DataType» Elevation
«Property» + elevationReference: ElevationReferenceValue + elevationValue: DirectPosition

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

«CodeList» ElevationReferenceValue	«CodeList» HeightStatusValue
+ aboveGroundEnvelope + bottomOfConstruction + entrancePoint + generalEave + generalGround + generalRoof + generalRoofEdge + highestEave + highestGroundPoint + highestPoint + highestRoofEdge + lowestEave + lowestFloorAboveGround + lowestGroundPoint + lowestRoofEdge + topOfConstruction	+ estimated + measured

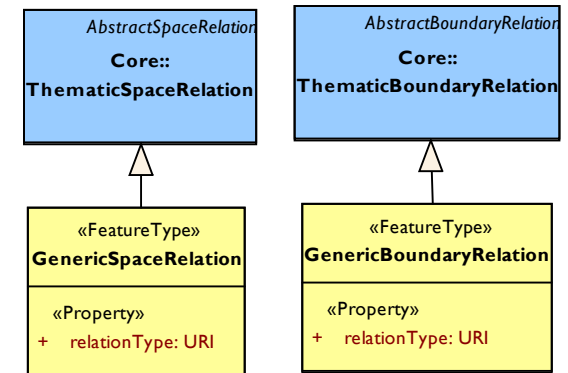
It needs to be discussed further within the SWG, whether a **Door** is a **Void** or an **Opening**,



Generics module

old

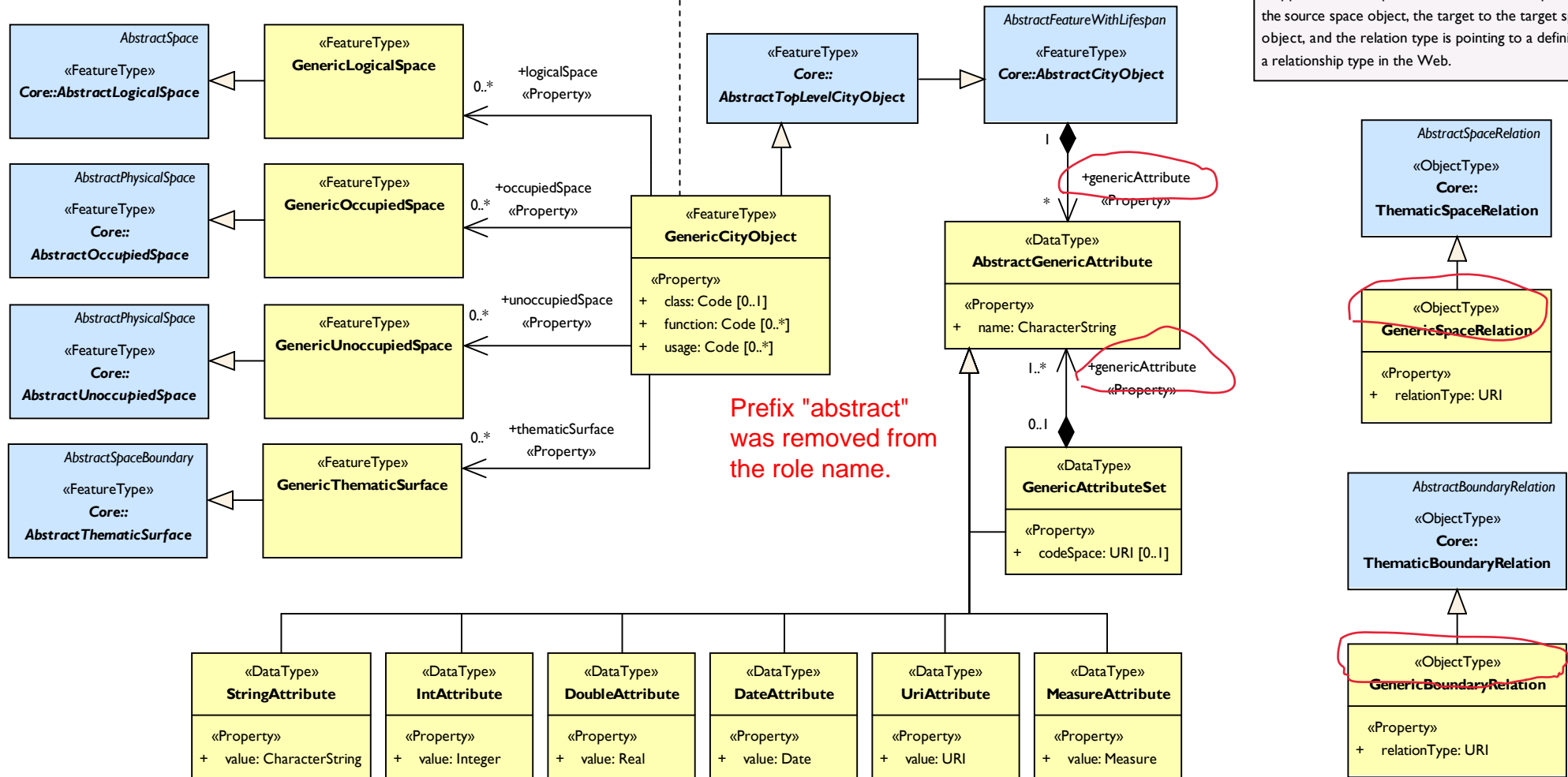
The **GenericSpaceRelation** and **GenericBoundaryRelation** can be used to represent an arbitrary relationship between two spaces or two space boundaries respectively. The relation type is expressed by an URI. Such relations could directly be mapped to RDF triples, where the source is pointing to the 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.



Generics module new

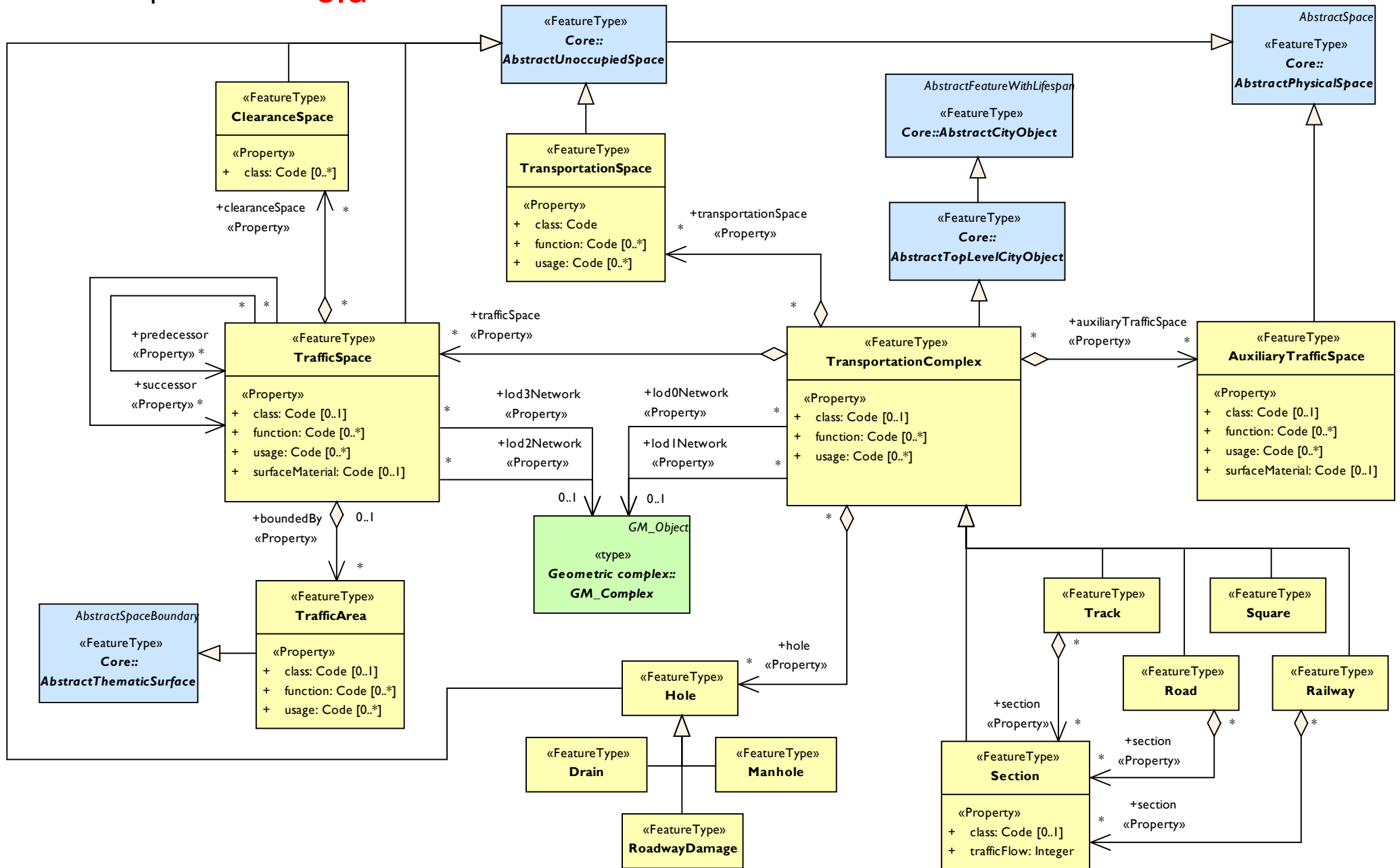
An instance of **GenericCityObject** can only be associated either with one or several instances of one of the Space classes or with one or several instances of **GenericThematicSurface**.

The **GenericSpaceRelation** and **GenericBoundaryRelation** can be used to represent an arbitrary relationship between two spaces or two space boundaries respectively. 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.



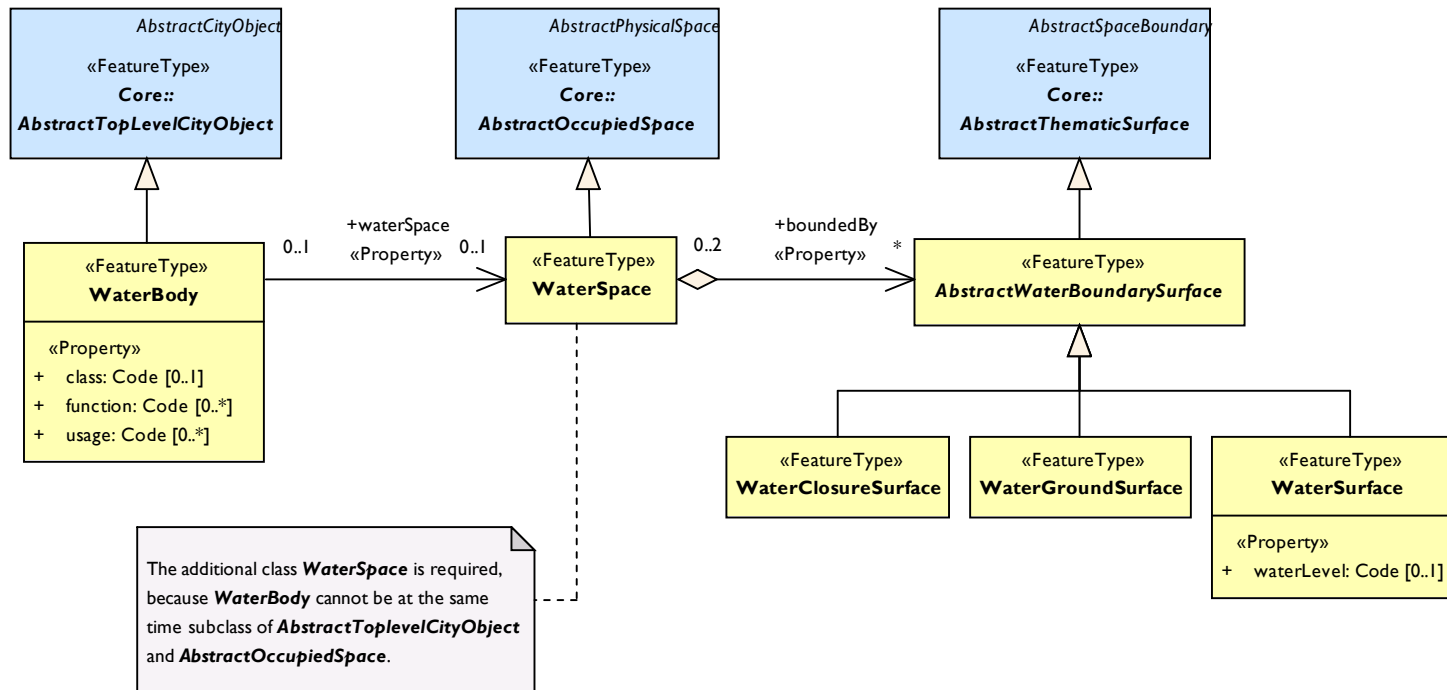
The stereotype **<<ObjectType>>** is now used instead of **<<FeatureType>>**, to be compliant with the stereotype of the supertype.

Transportation module old



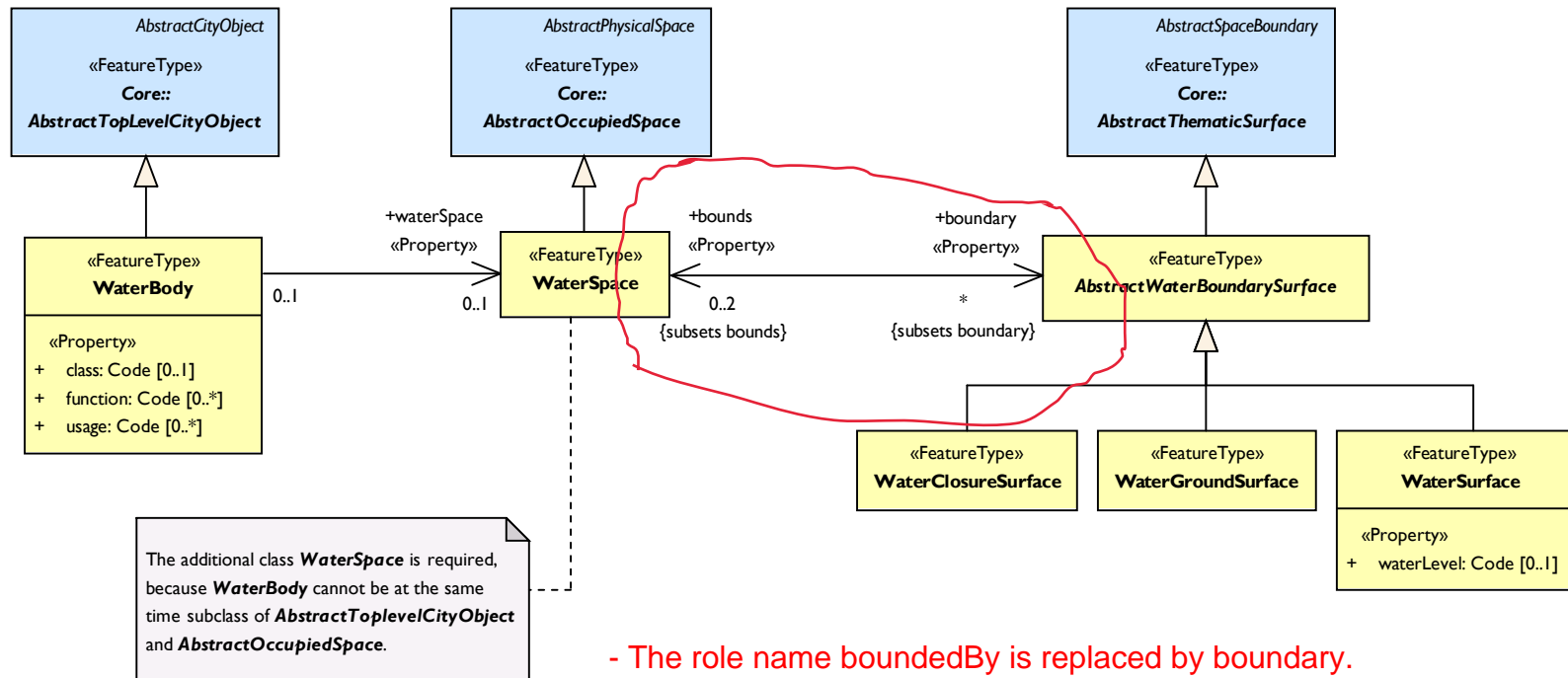
WaterBody module

old



WaterBody module

new



- The role name **boundedBy** is replaced by **boundary**.
- The subsets relation is represented through property strings **{subsets ...}**.
- The association is represented bidirectionally, i.e. identically to the association in the Core module.