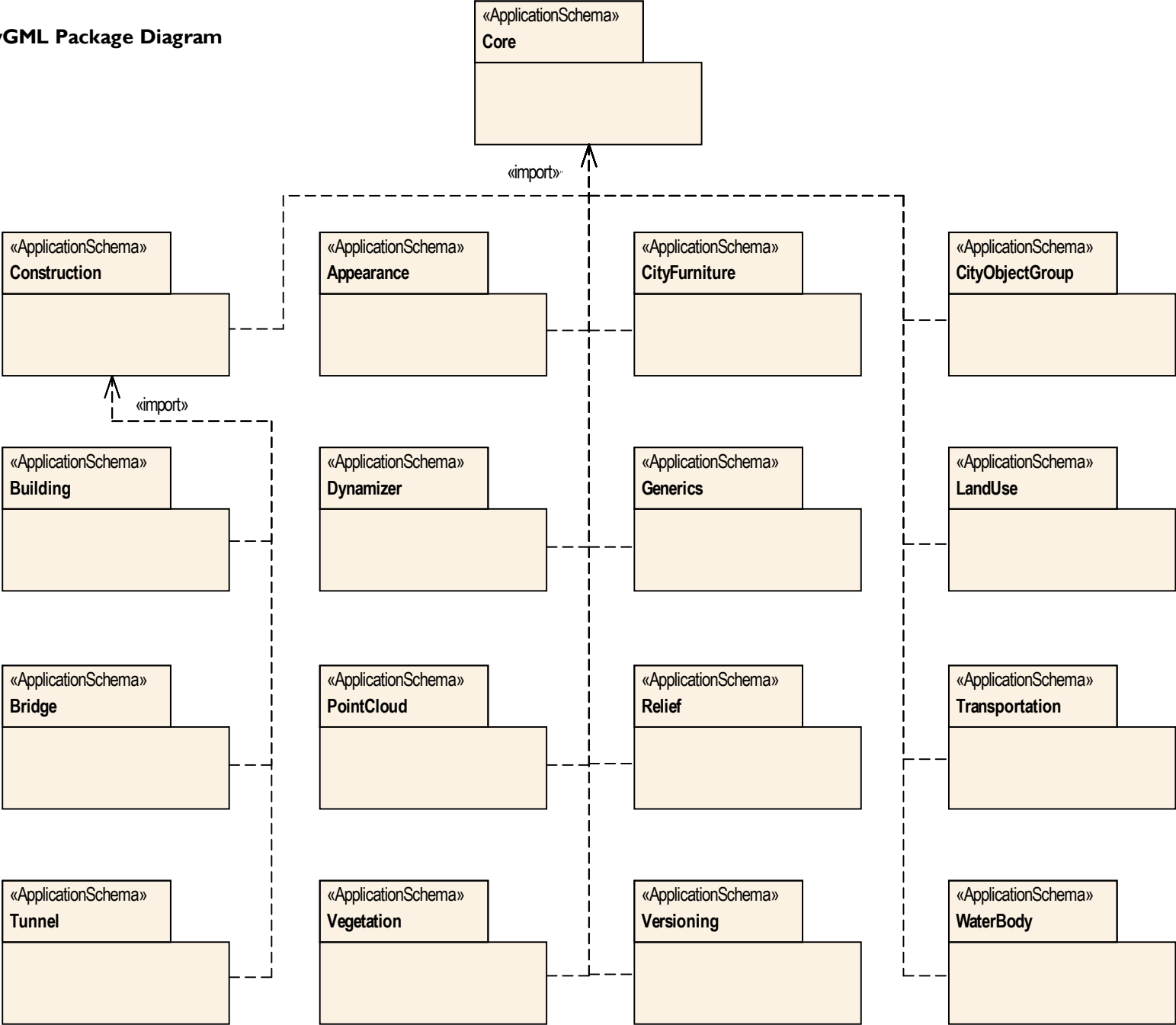
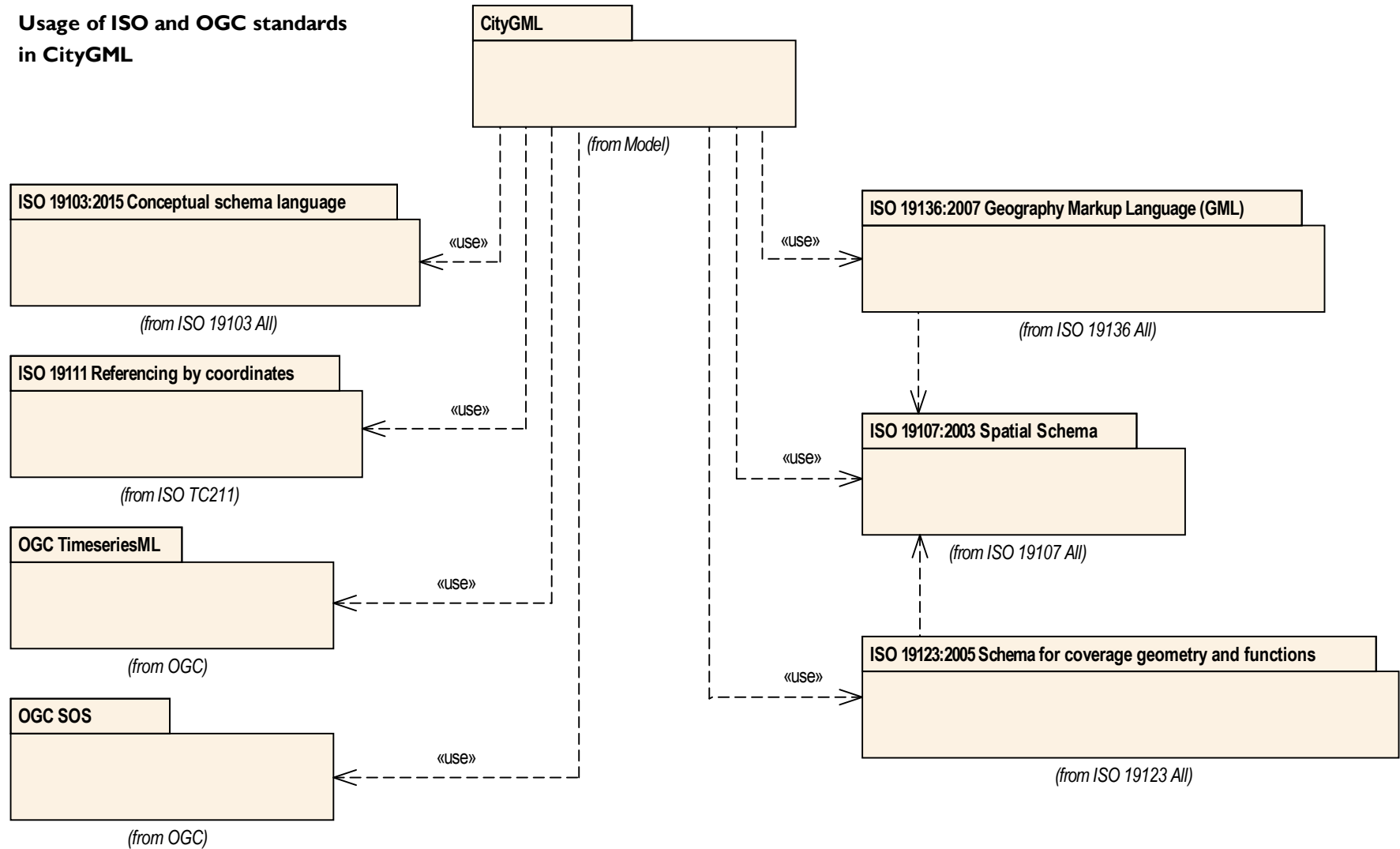


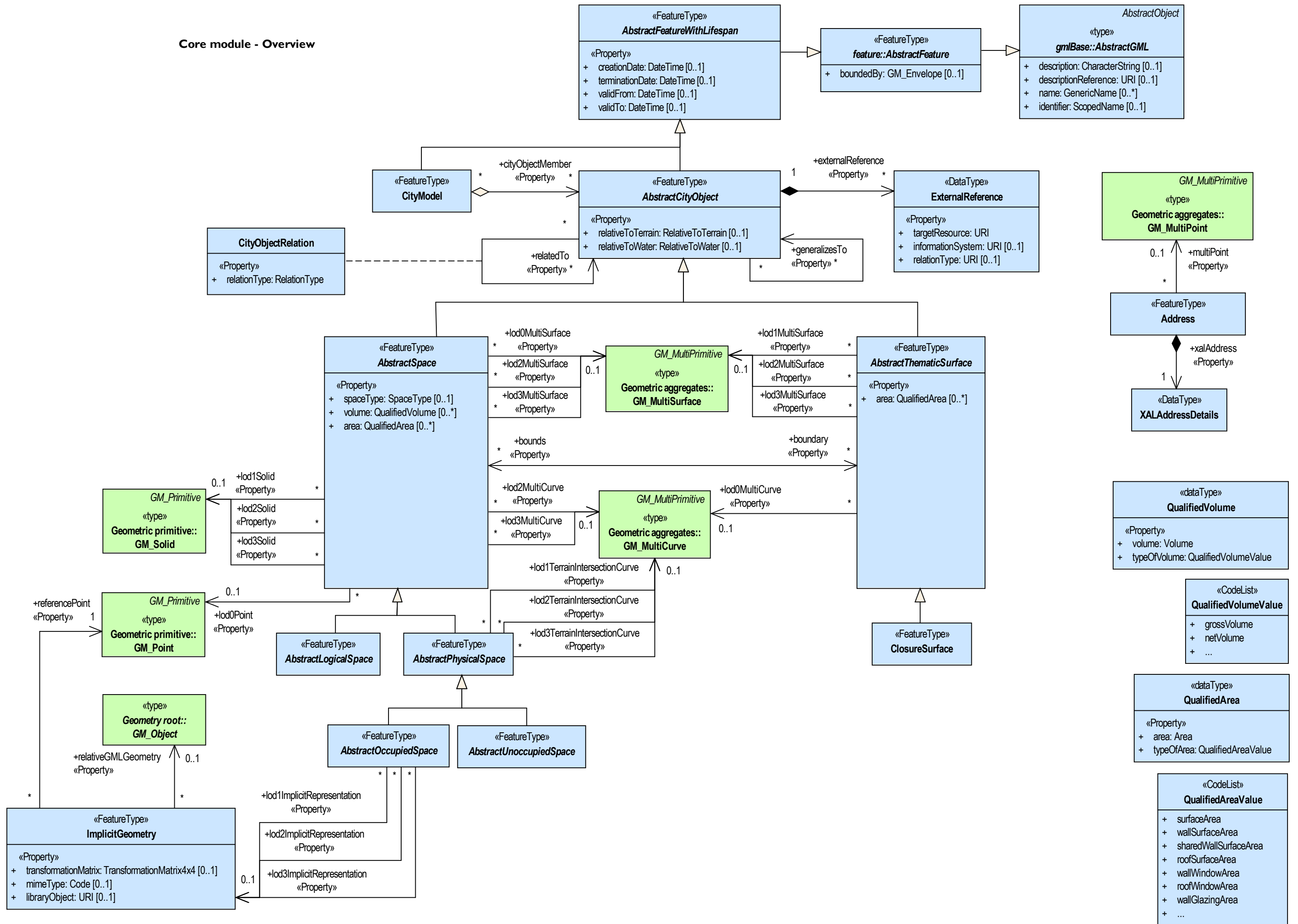
CityGML Package Diagram



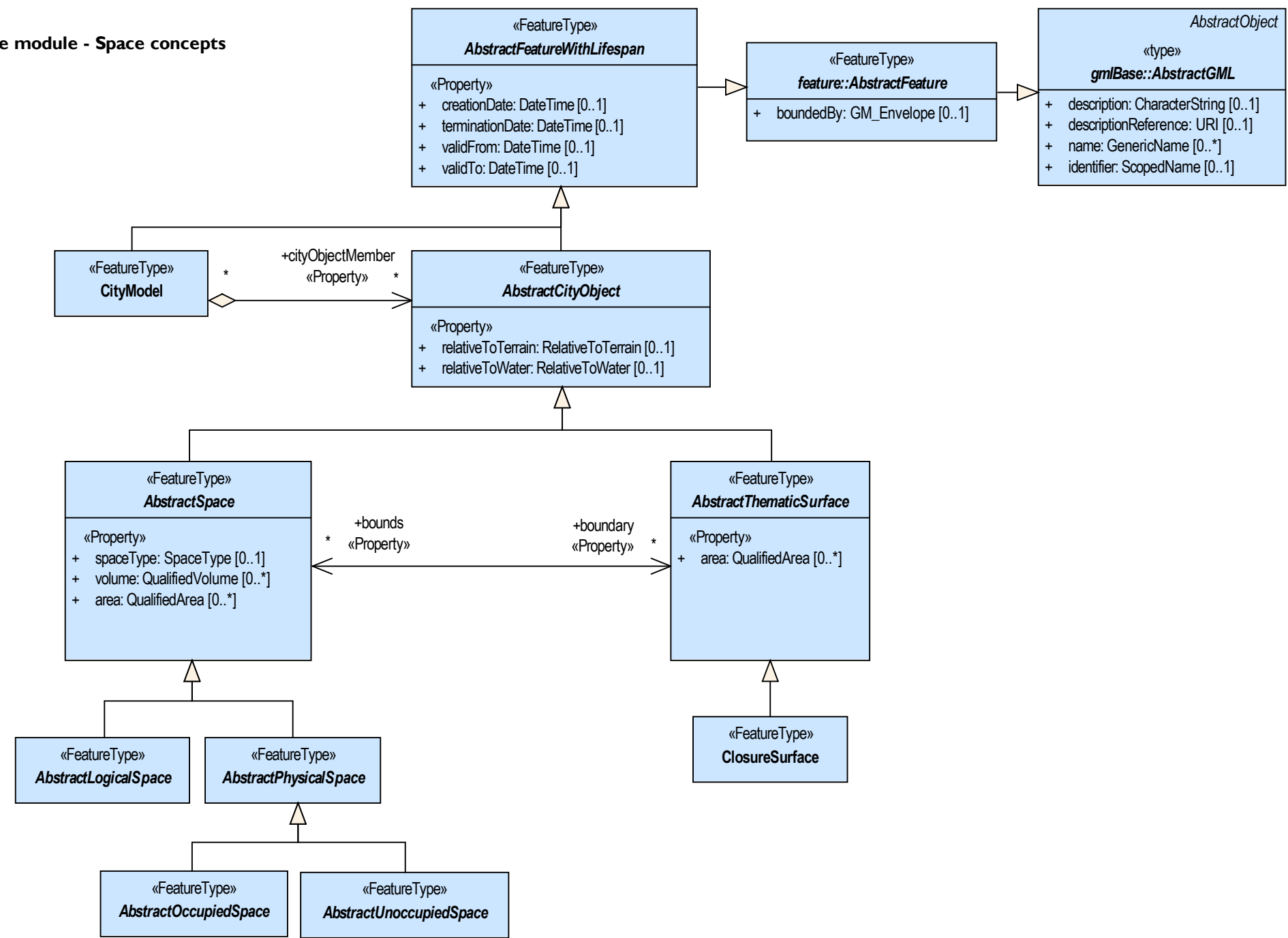
Usage of ISO and OGC standards in CityGML



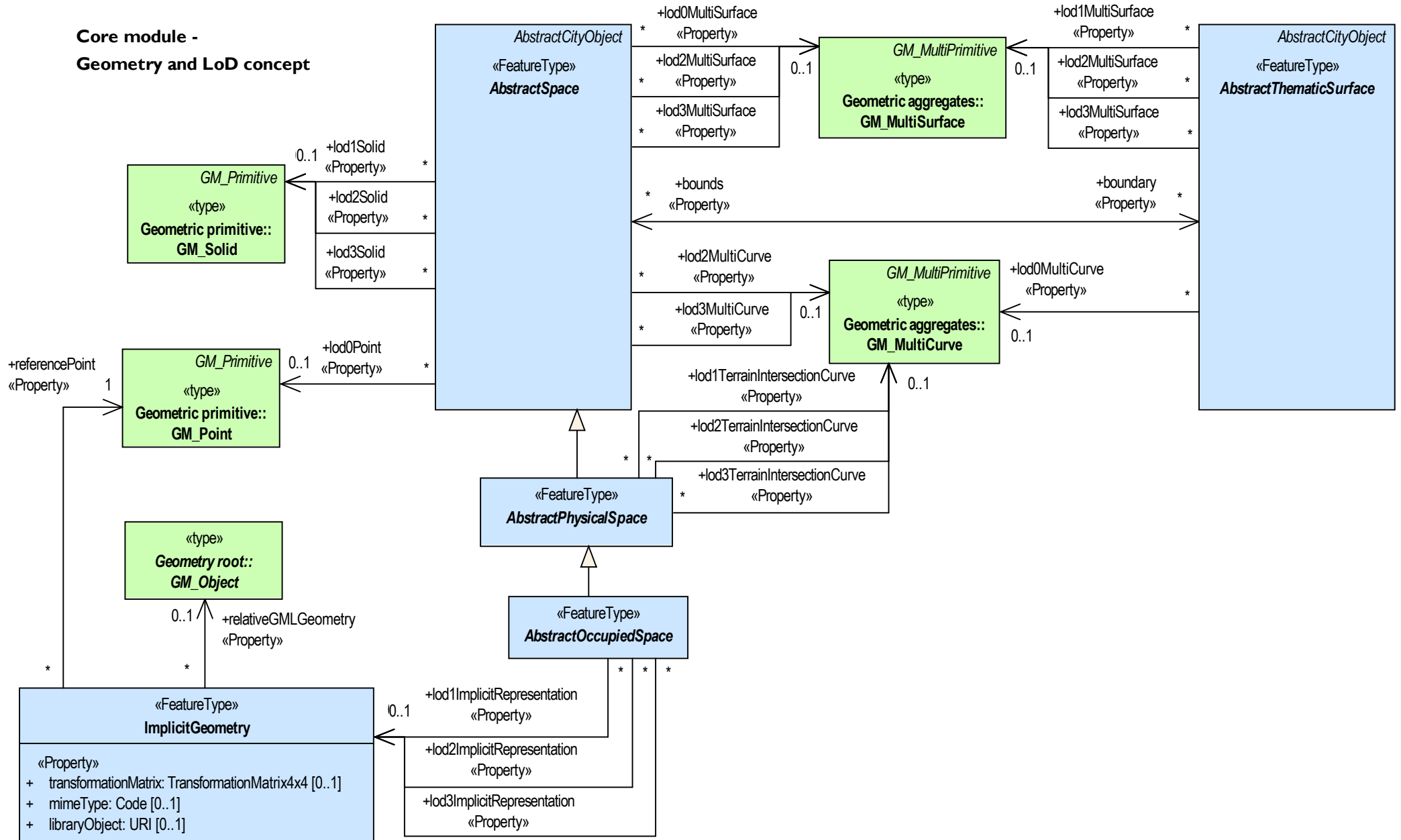
Core module - Overview



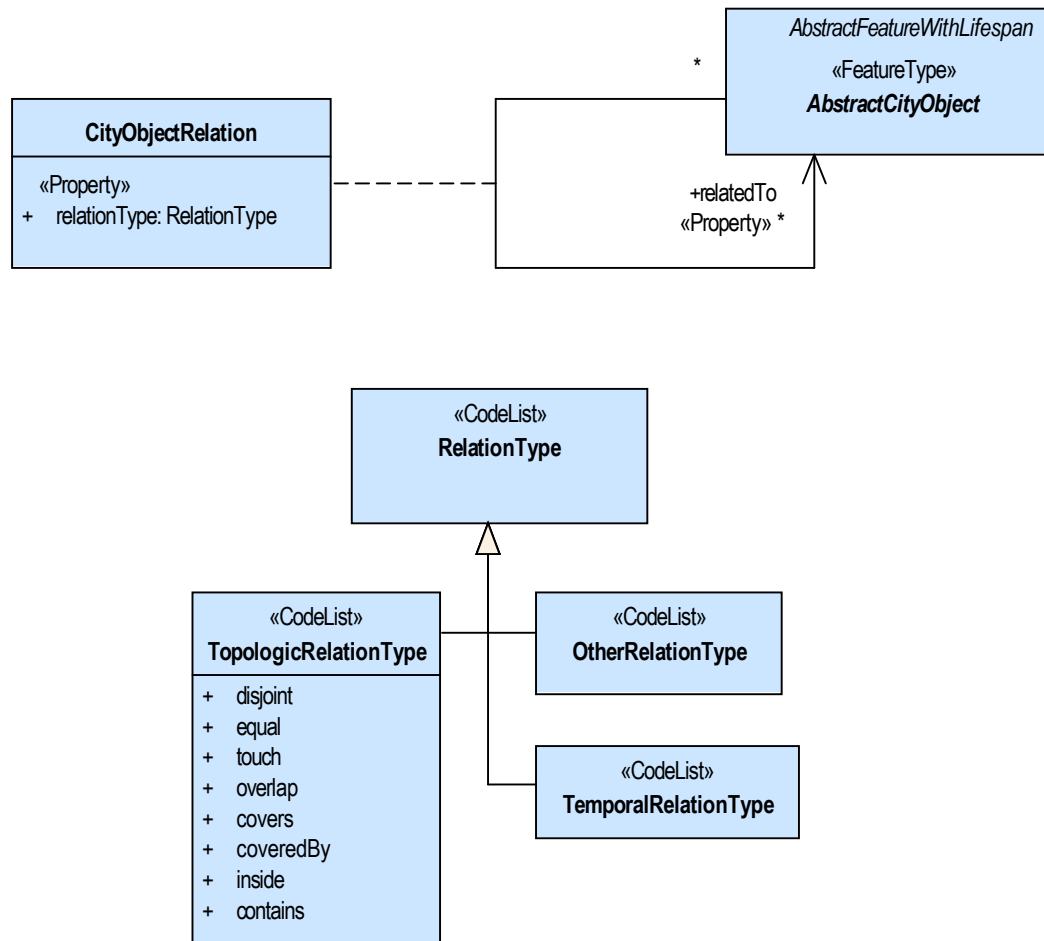
Core module - Space concepts



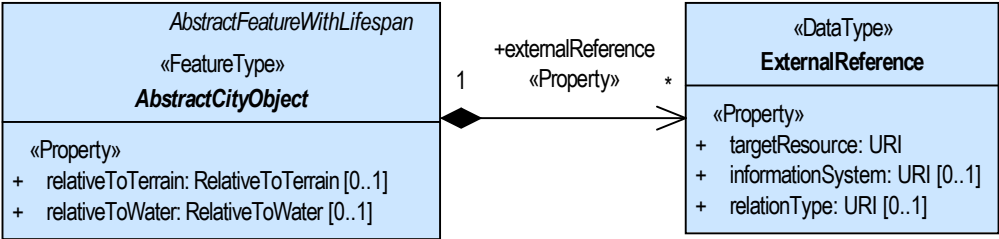
**Core module -
Geometry and LoD concept**



Core module - City object relations



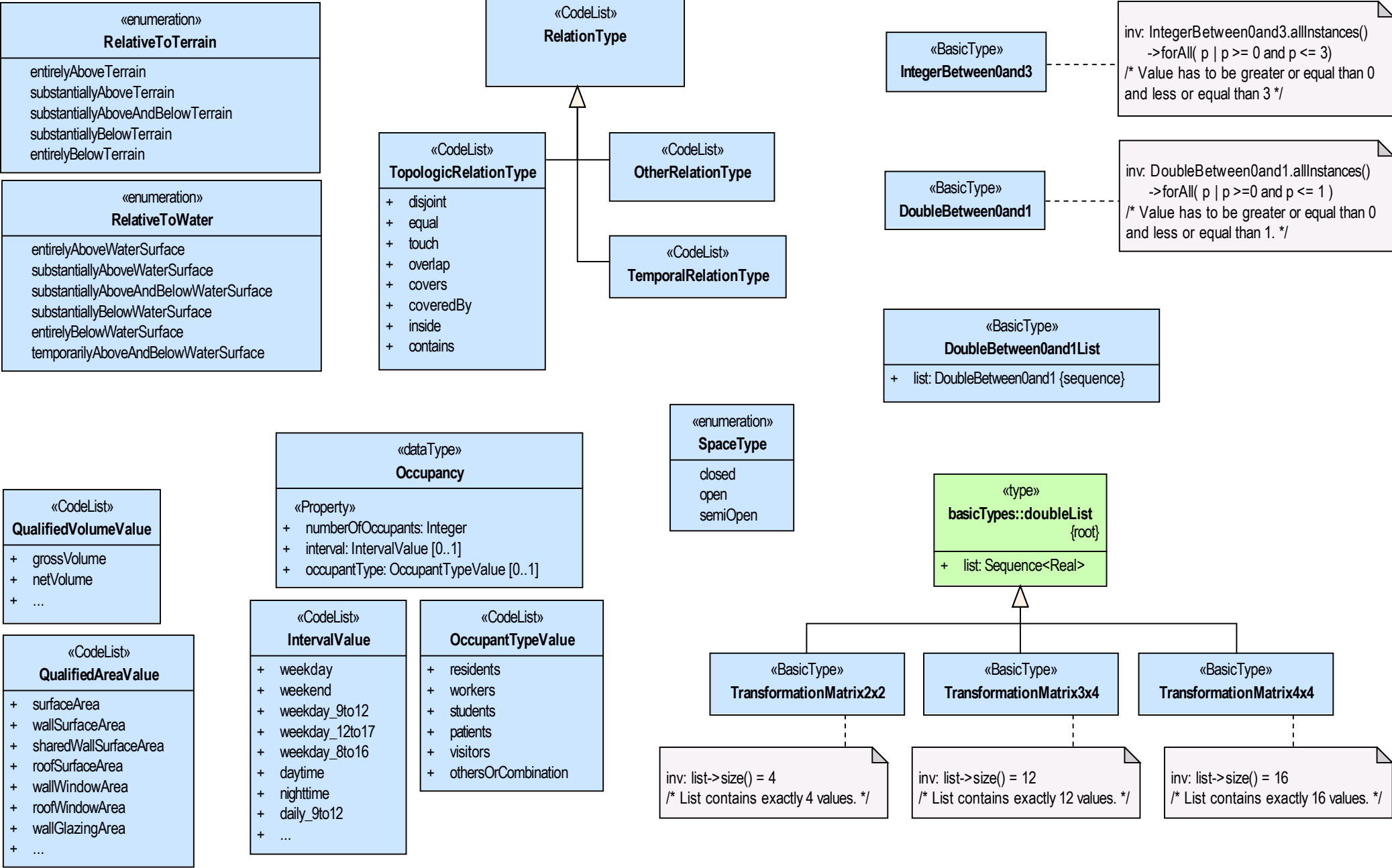
Core module - Miscellaneous

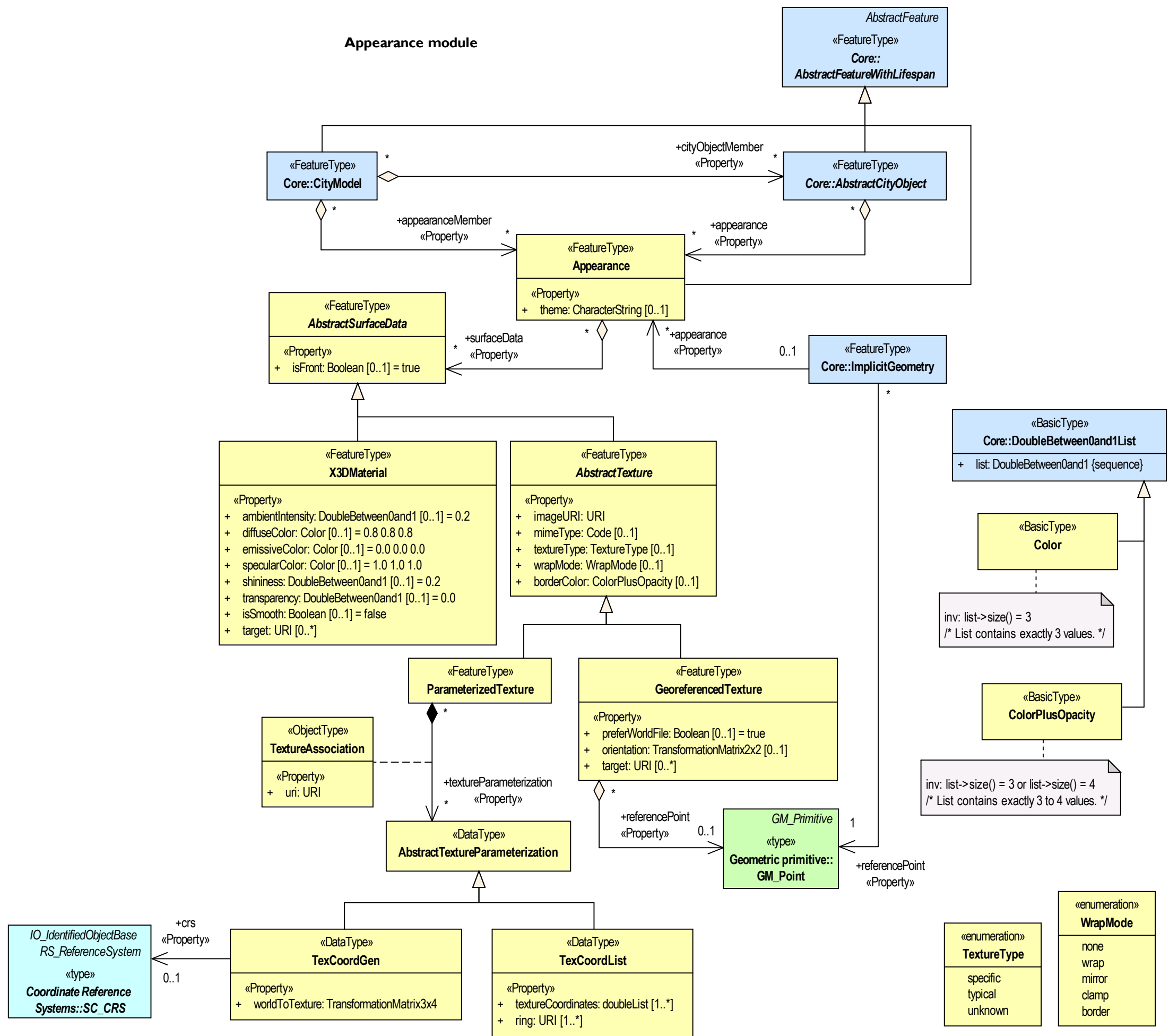


ExternalReference is now extended by an optional **relation Type** which can link to some external definition of the type of relation (e.g. the *sameAs* relation from OWL). Hence, **ExternalReferences** can now be used to express relations similar to RDF.

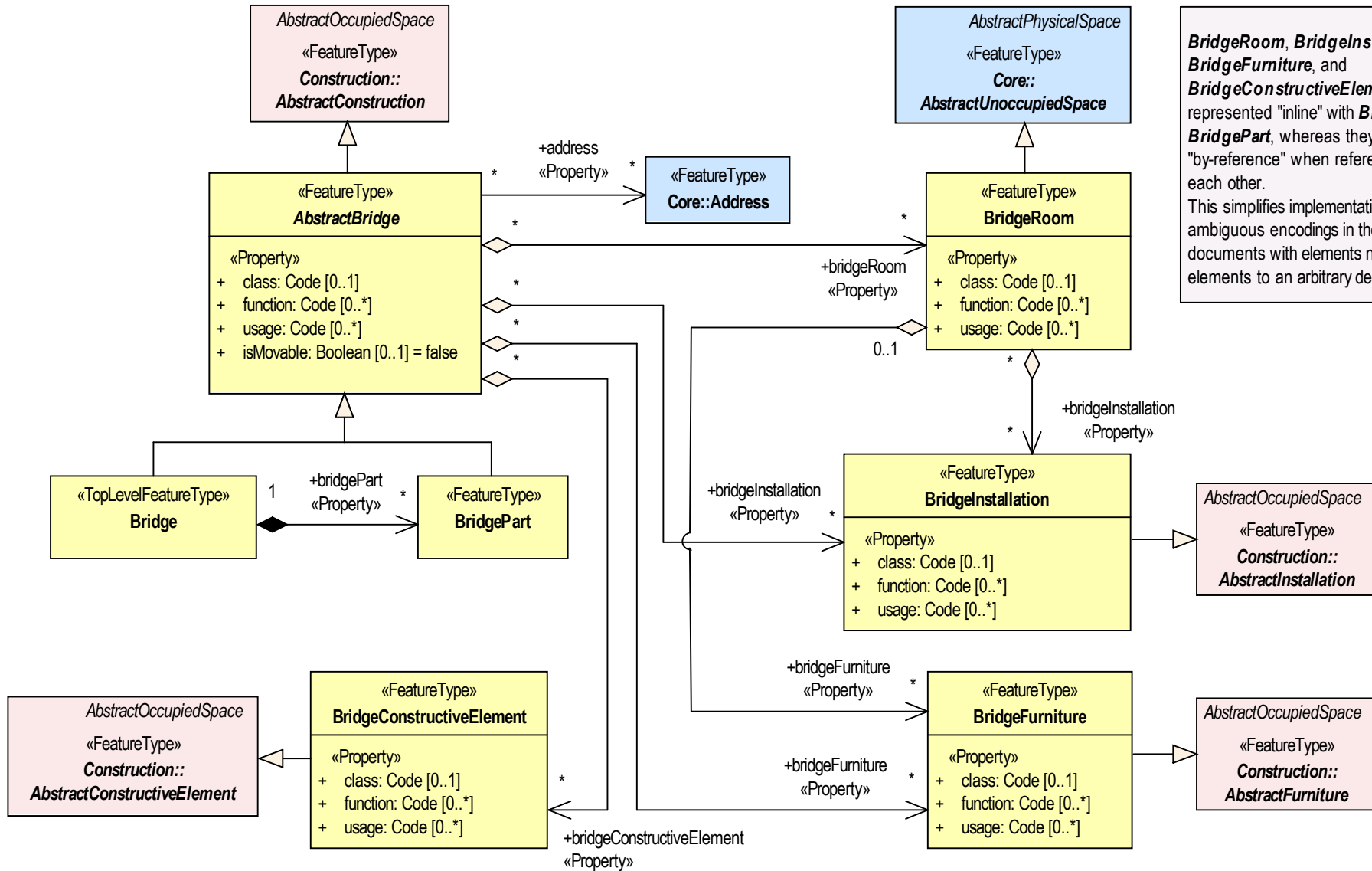


Core module - Basic Types and Enumerations

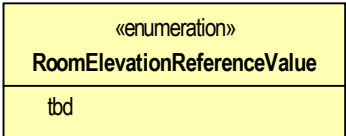
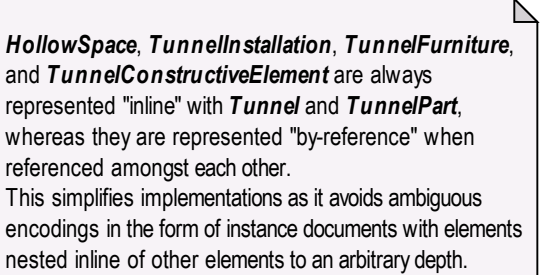




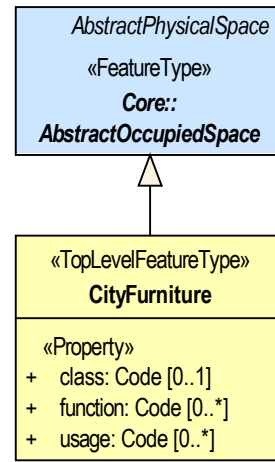
Bridge module



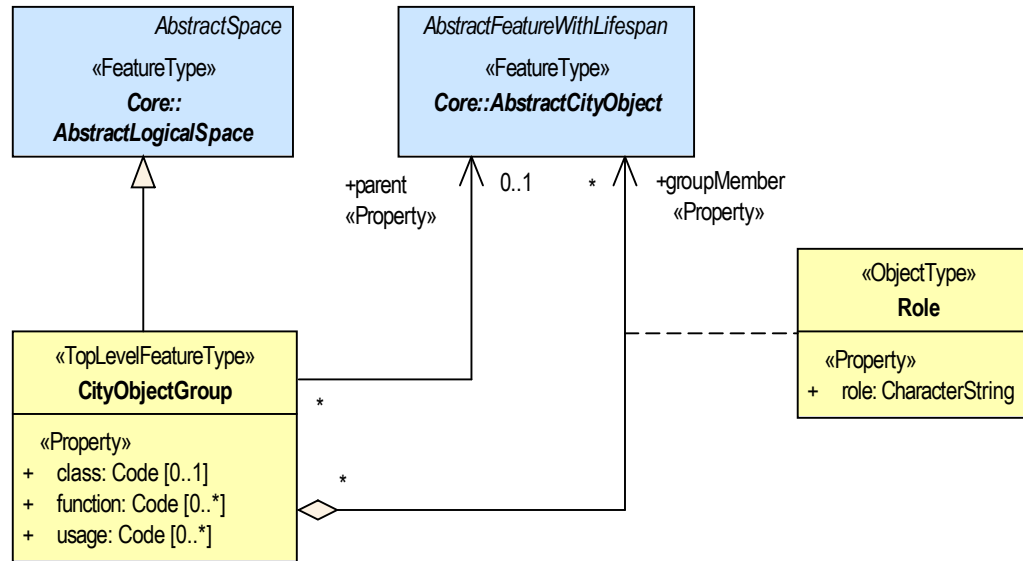
BridgeRoom, **BridgeInstallation**, **BridgeFurniture**, and **BridgeConstructiveElement** are always represented "inline" with **Bridge** and **BridgePart**, whereas they are represented "by-reference" when referenced amongst each other. This simplifies implementations as it avoids ambiguous encodings in the form of instance documents with elements nested inline of other elements to an arbitrary depth.

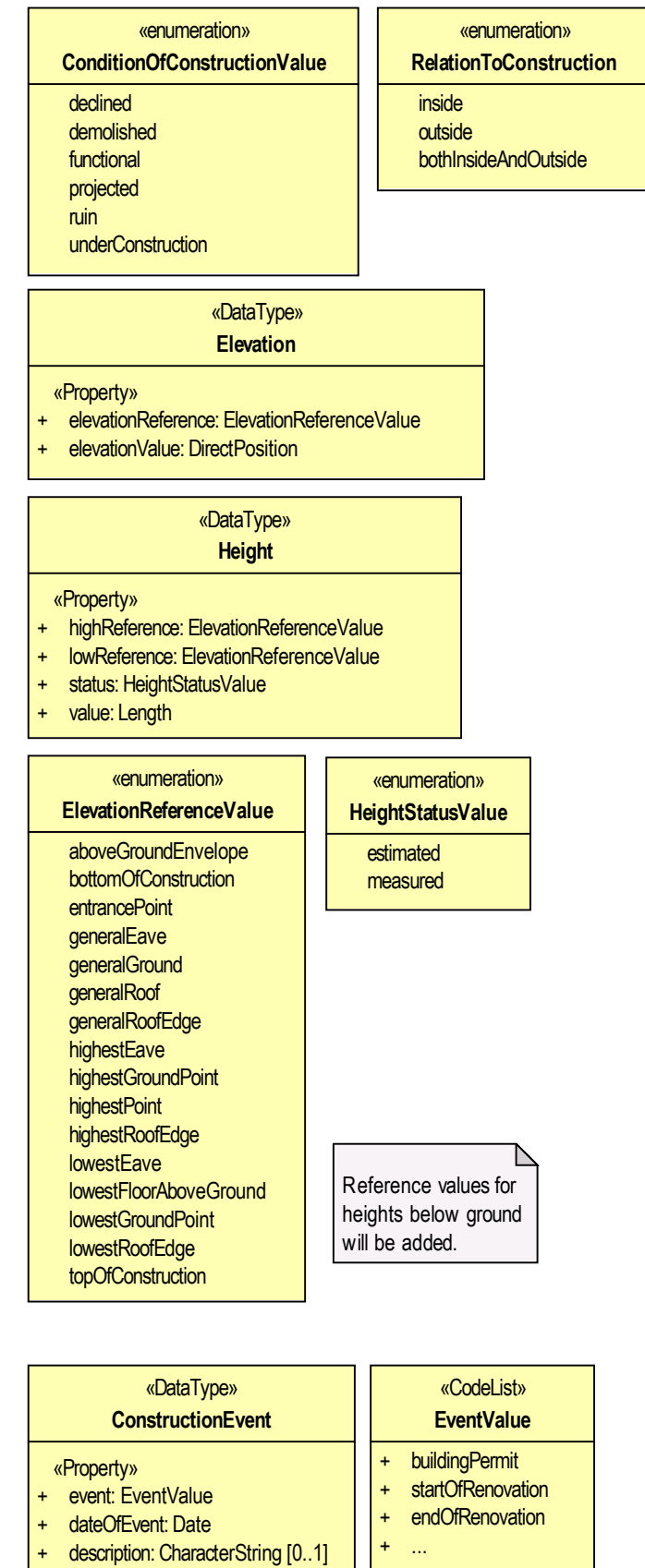
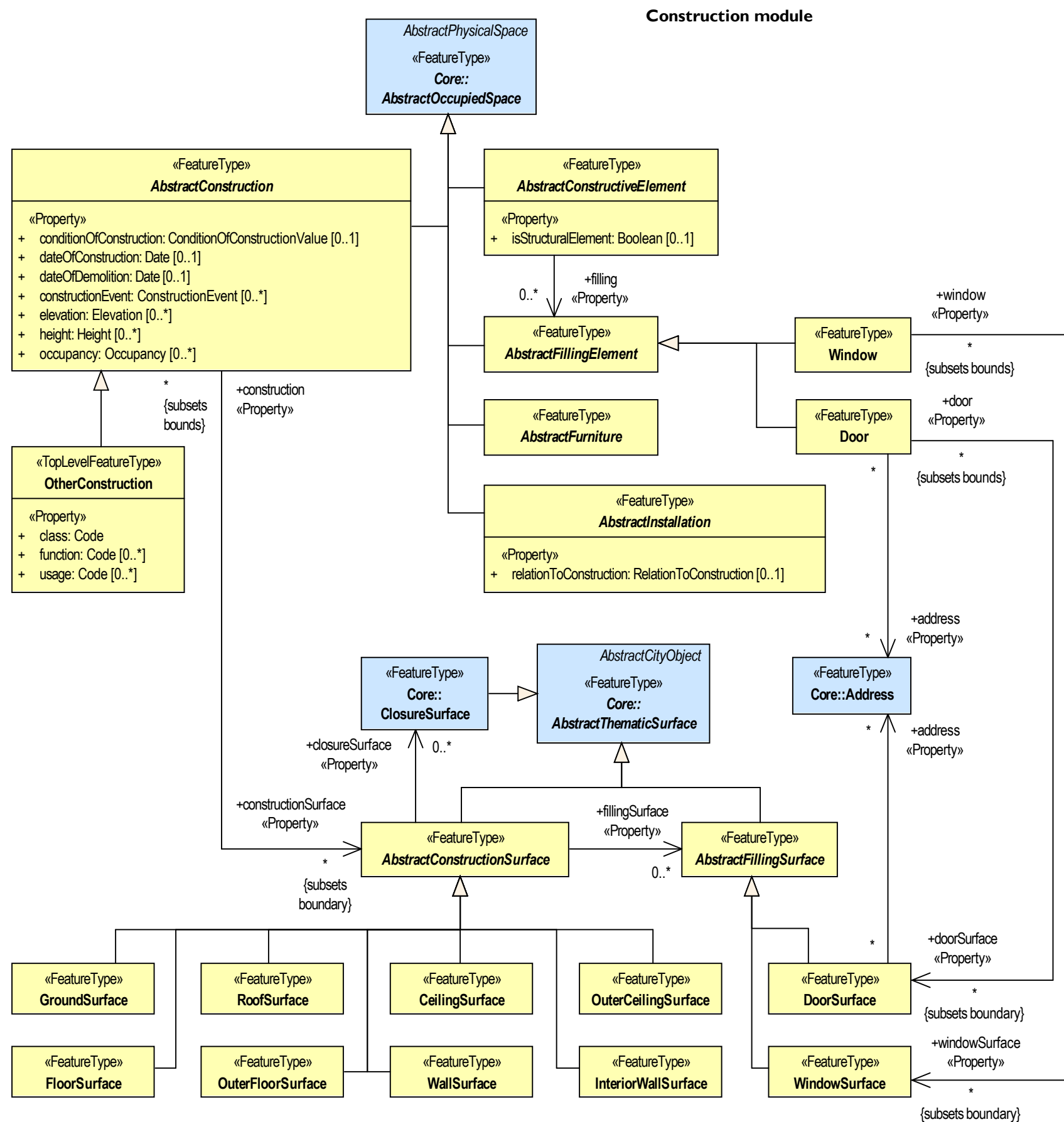


CityFurniture module

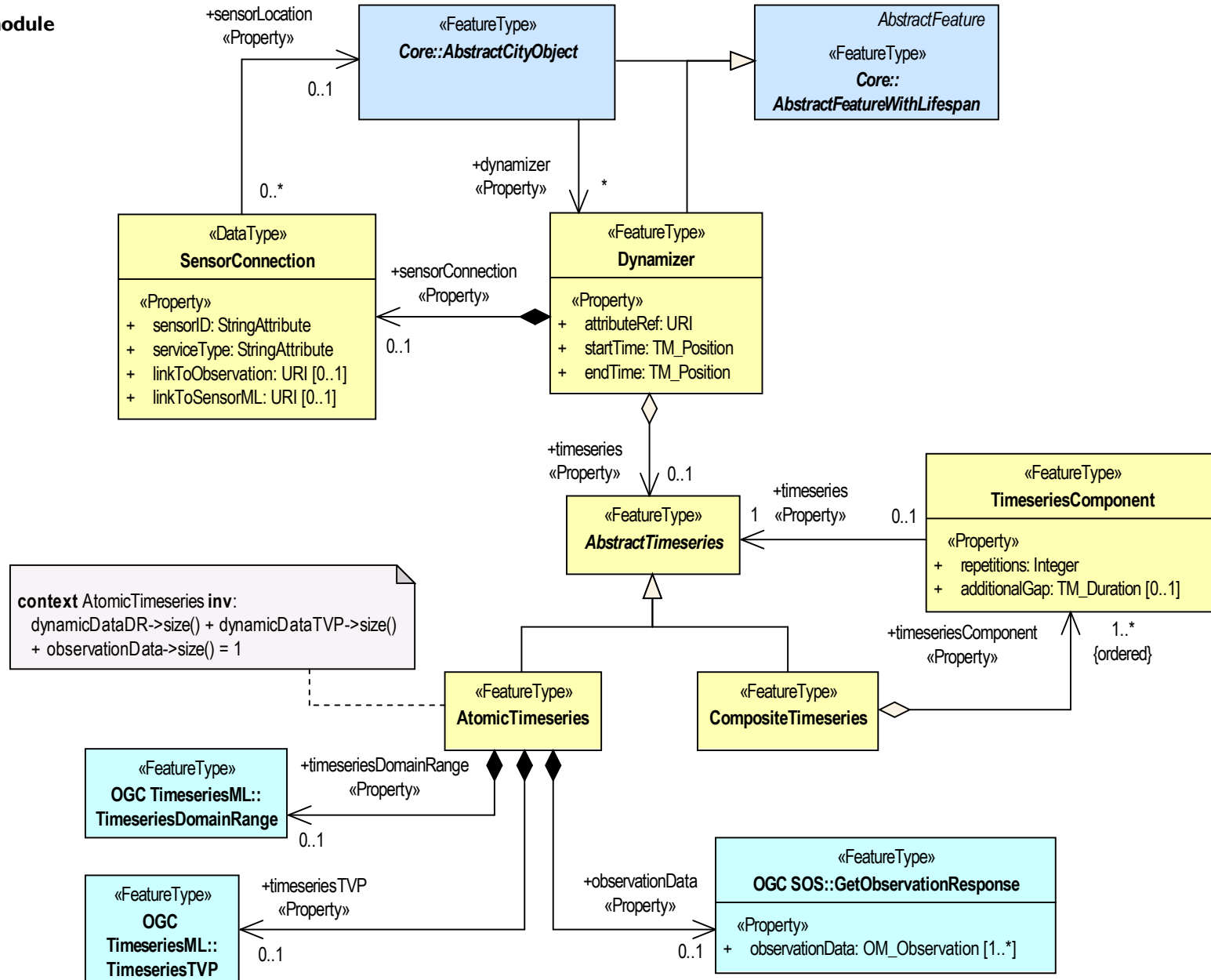


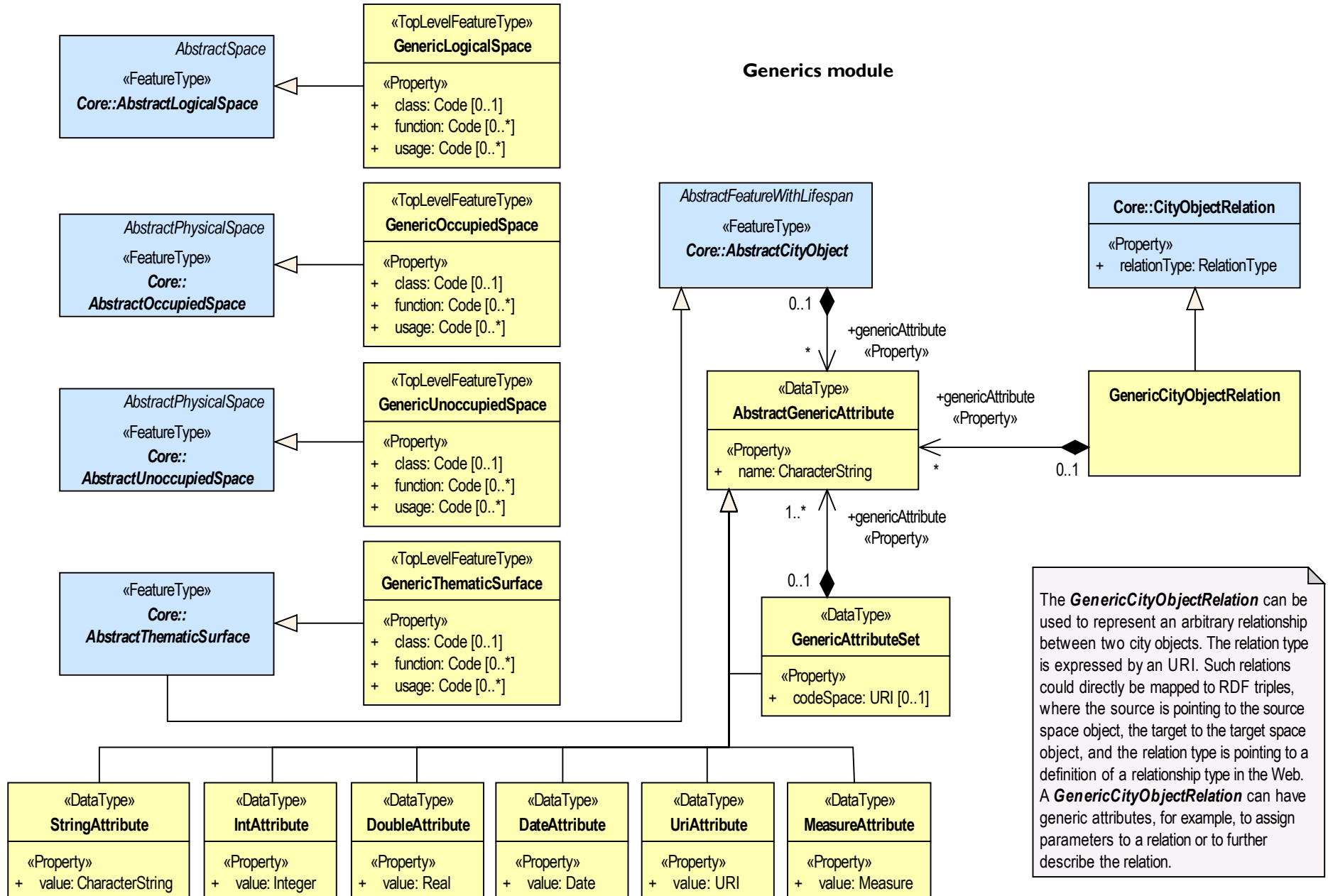
CityObjectGroup module





Dynamizer module

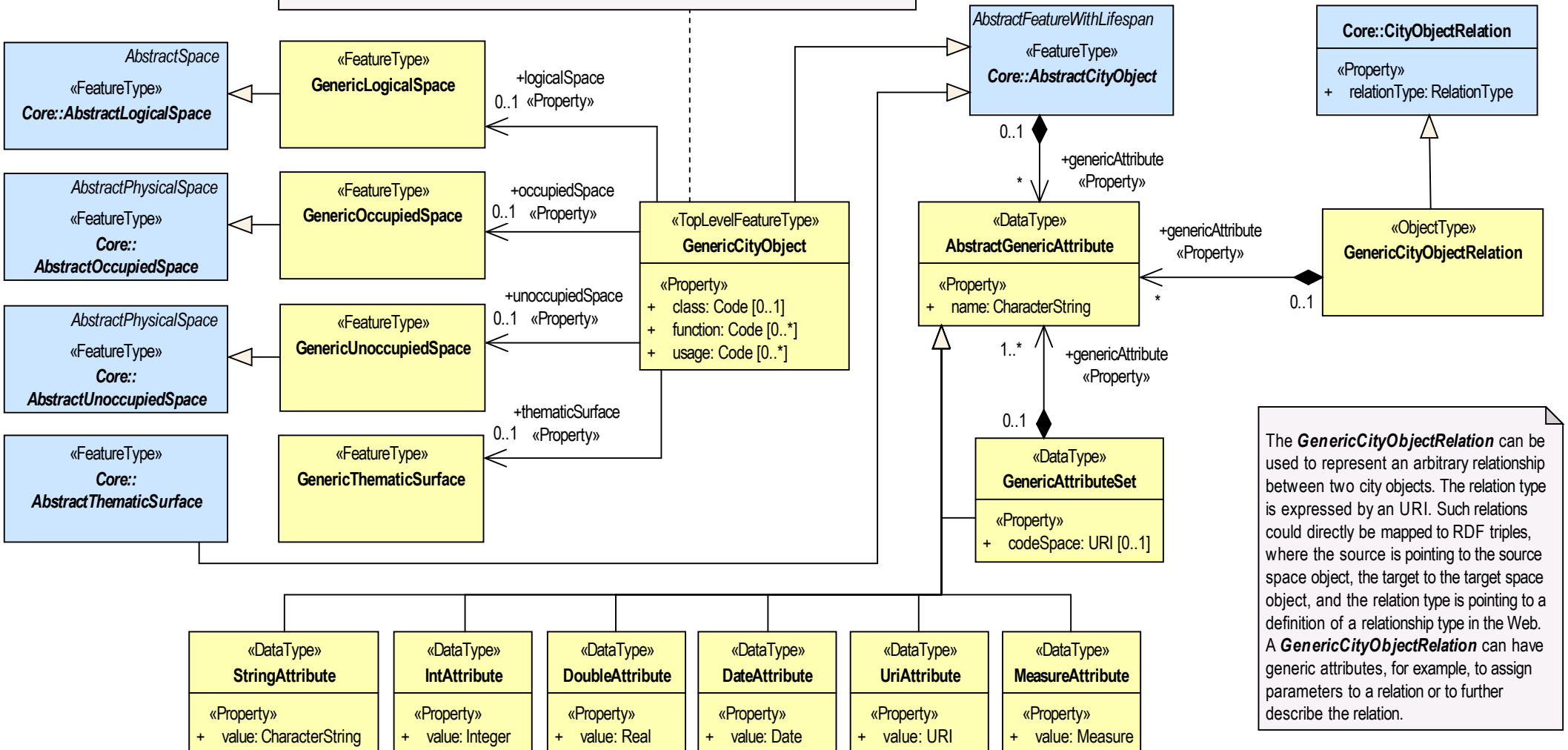




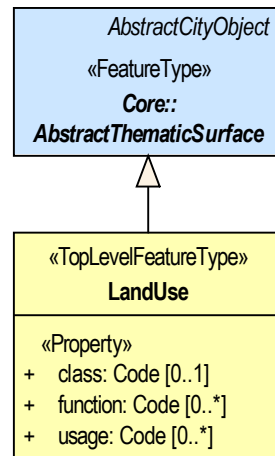
Generics module - Alternative I

context GenericCityObject inv:
 logicalSpace->size() + occupiedSpace->size()
 + unoccupiedSpace->size() + thematicSurface->size() = 1

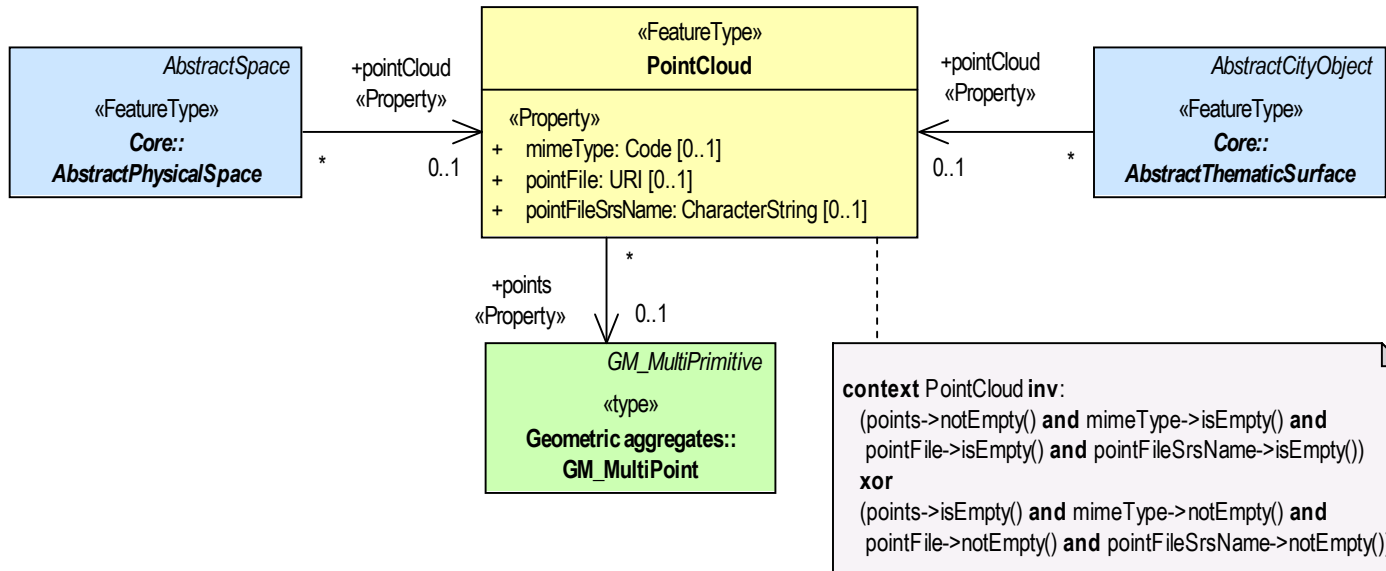
An instance of **GenericCityObject** can only be associated either with one instance of one of the Space classes or with one instance of **GenericThematicSurface**.
 If several city objects are to be modelled at the same time, CityObjectGroup should be used.



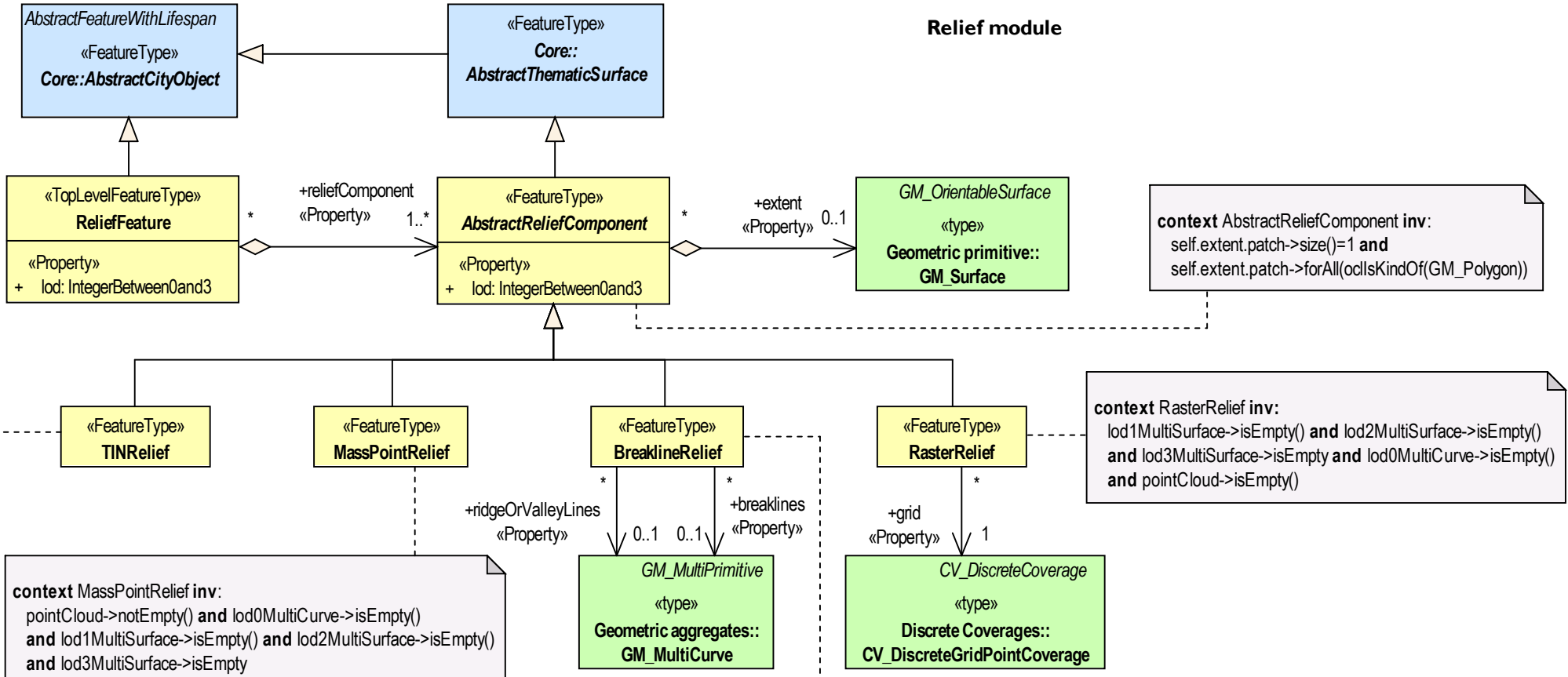
LandUse module



PointCloud module



Relief module



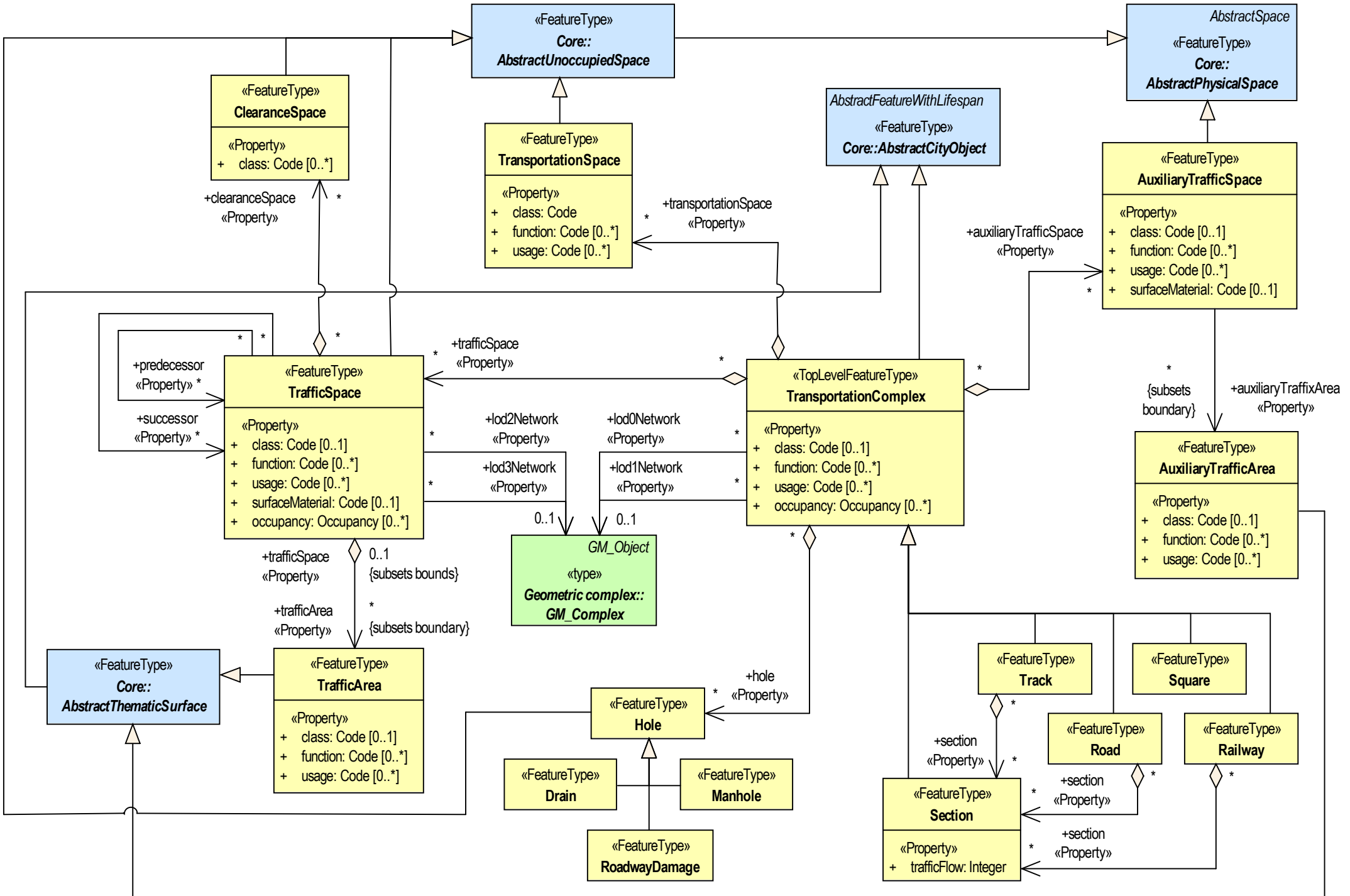
context TINRelief inv:
 lod0MultiCurve->isEmpty() and pointCloud->isEmpty()
 and if lod = 1
 then lod1MultiSurface->notEmpty() and lod1MultiSurface.isKindOf(GM_TriangulatedSurface)
 else if lod = 2
 then lod2MultiSurface->notEmpty() and lod2MultiSurface.isKindOf(GM_TriangulatedSurface)
 else lod = 3 and lod3MultiSurface->notEmpty() and lod3MultiSurface.isKindOf(GM_TriangulatedSurface)
 endif
 endif
)

This OCL constraint expresses that:

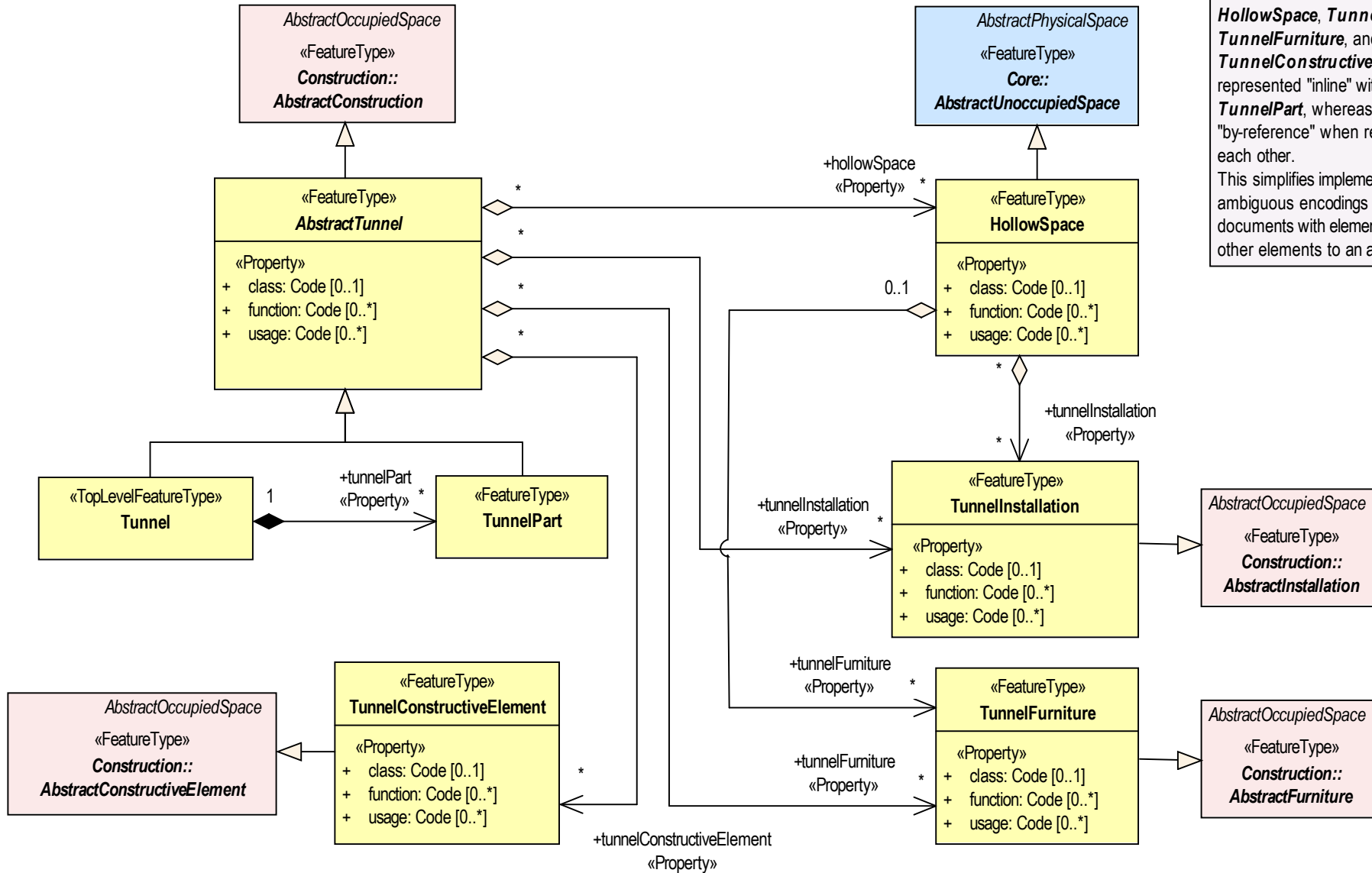
- 1) the properties "lod0MultiCurve" and "pointCloud" must not be used
- 2) and depending on the value of the property "lod" either the property "lod1MultiSurface", "lod2MultiSurface", or "lod3MultiSurface" must be used and be of type "GM_TriangulatedSurface" or of its subclass "GM_Tin"

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.



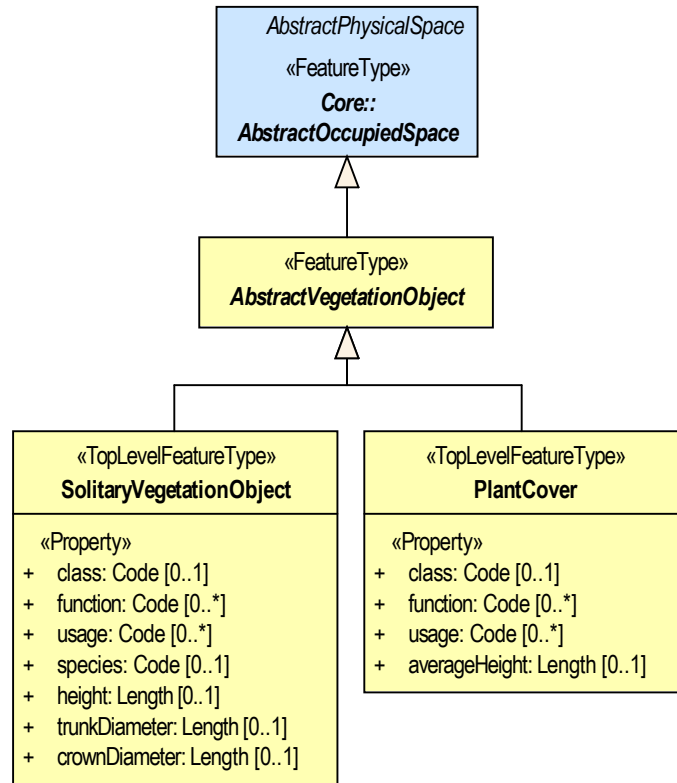
Tunnel module



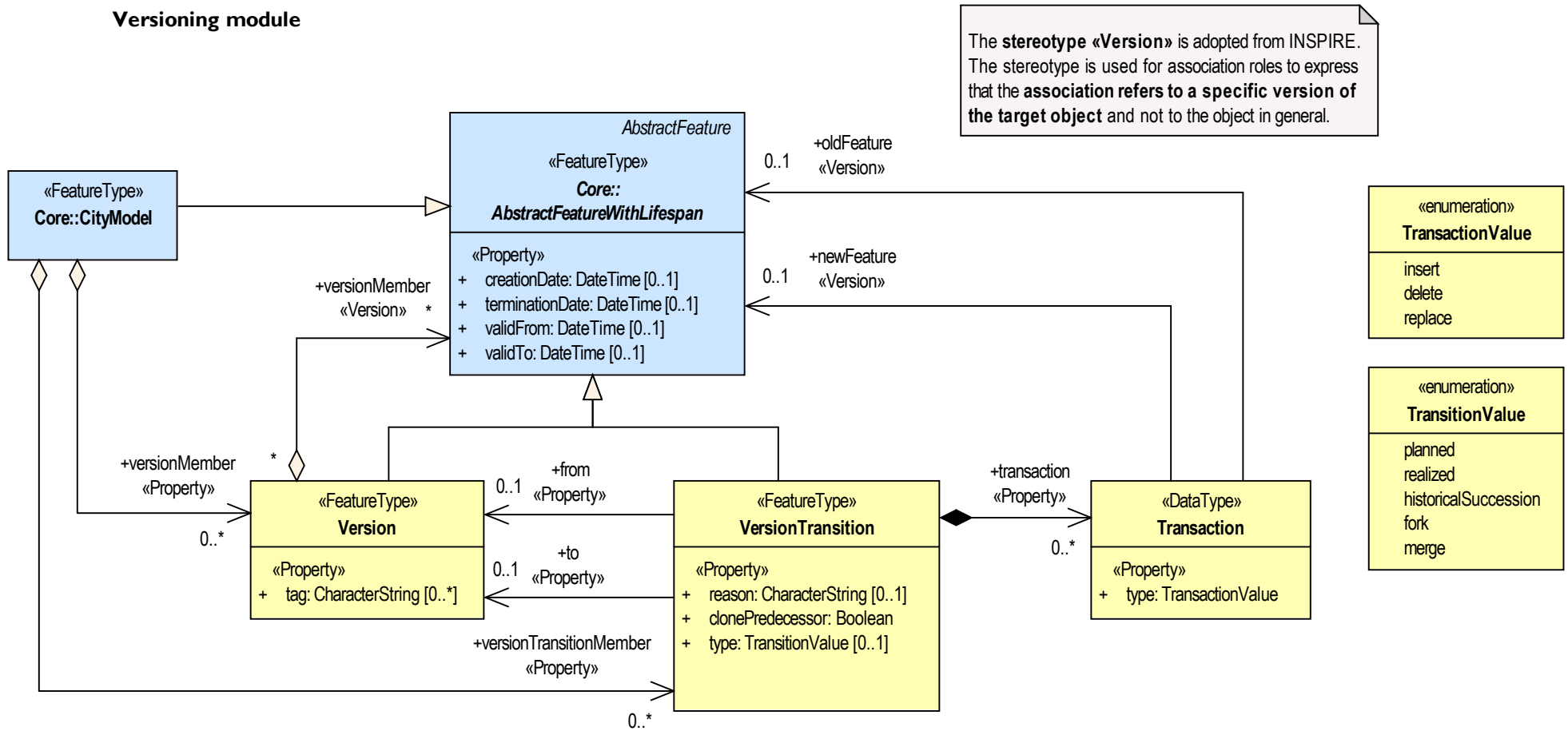
HollowSpace, **TunnelInstallation**, **TunnelFurniture**, and **TunnelConstructiveElement** are always represented "inline" with **Tunnel** and **TunnelPart**, whereas they are represented "by-reference" when referenced amongst each other.

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.

Vegetation module



Versioning module



WaterBody module

