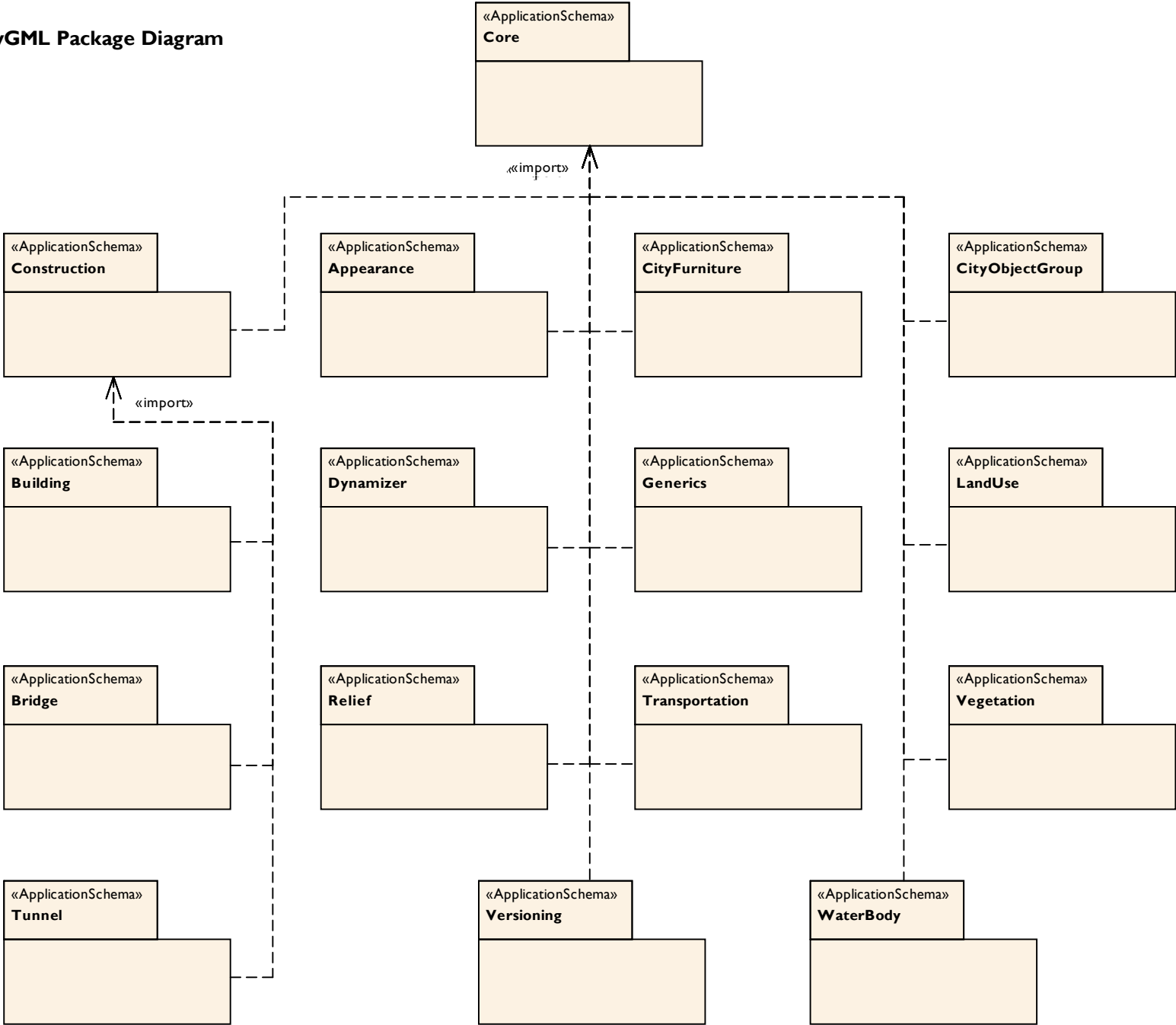
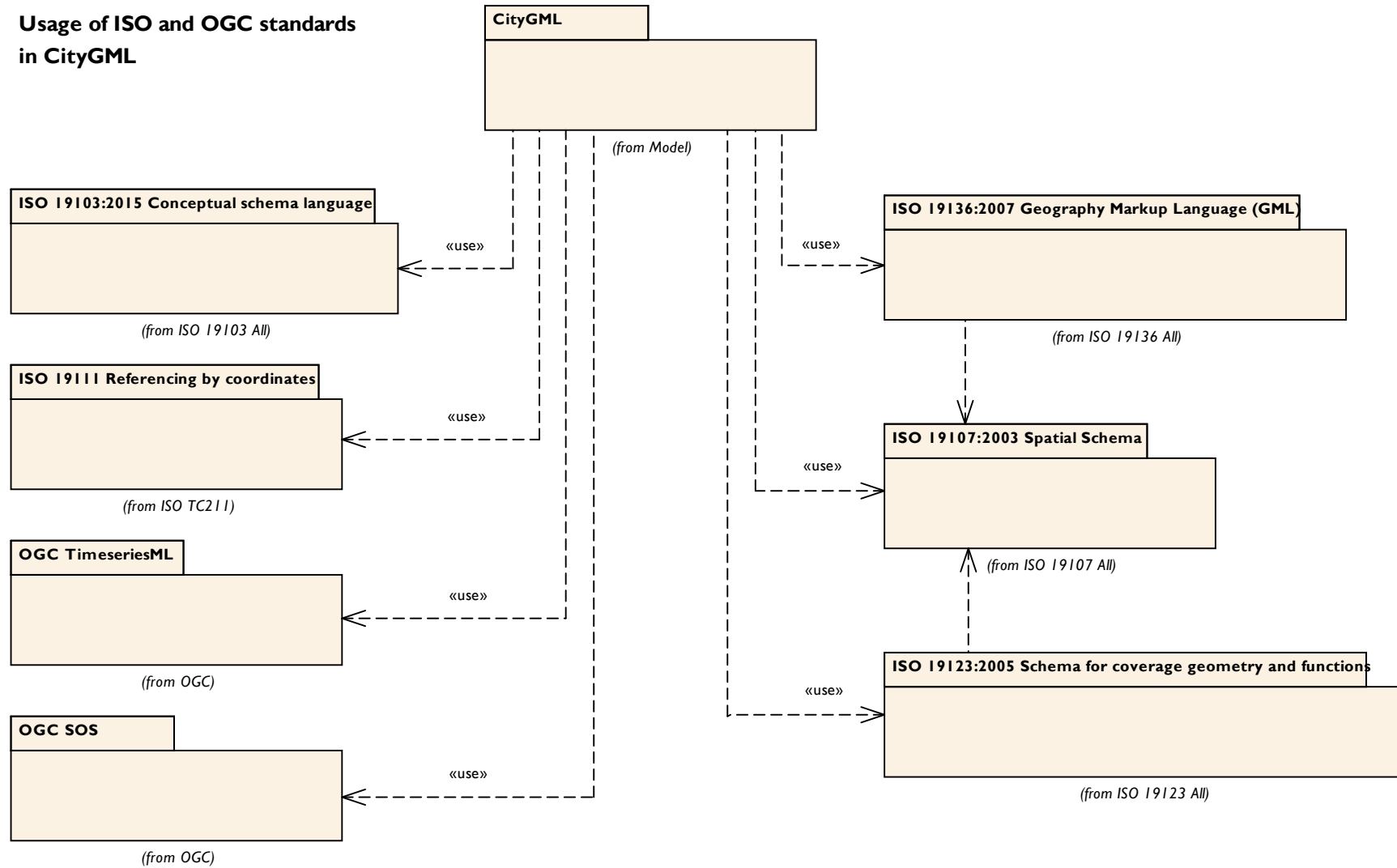


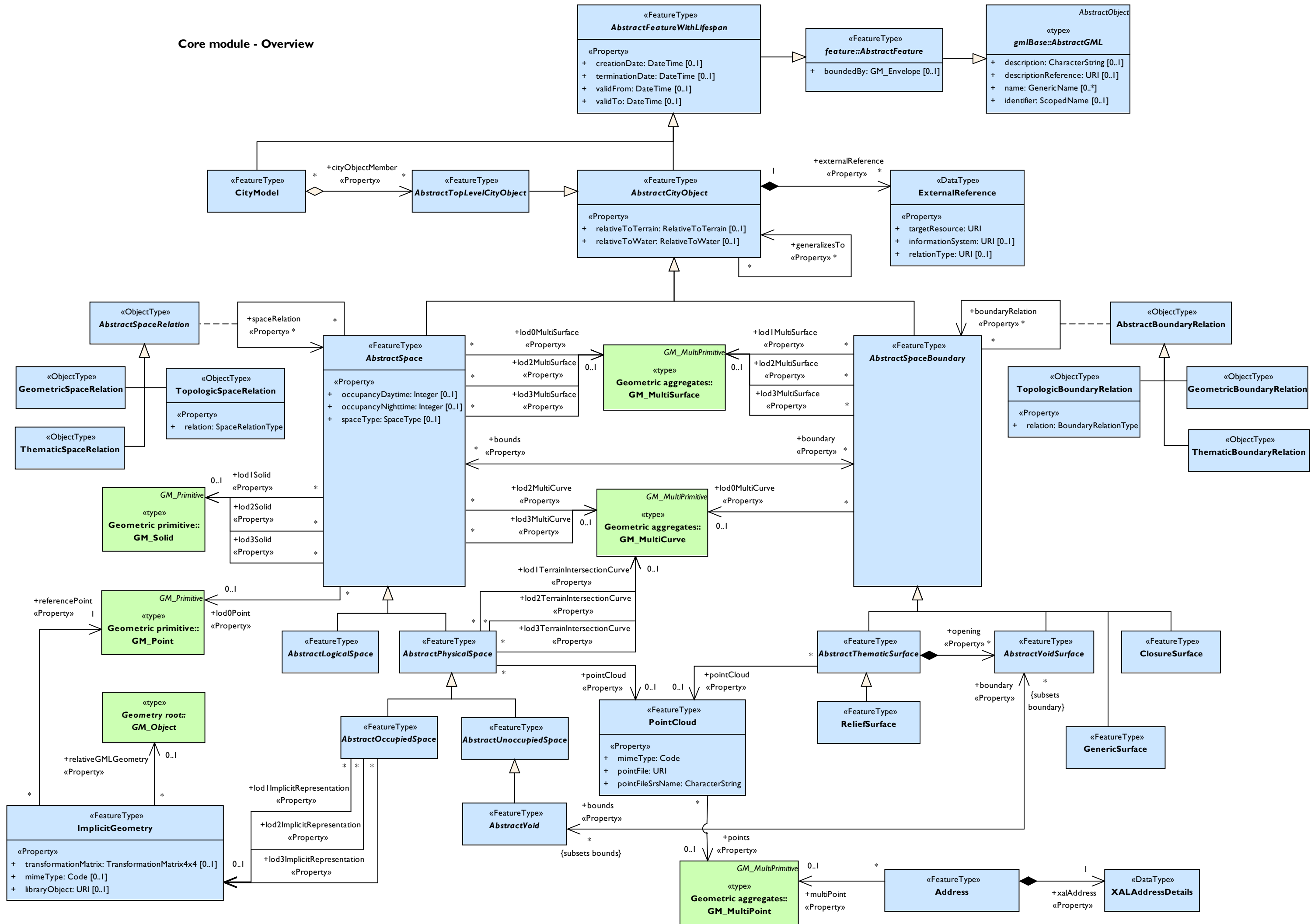
CityGML Package Diagram



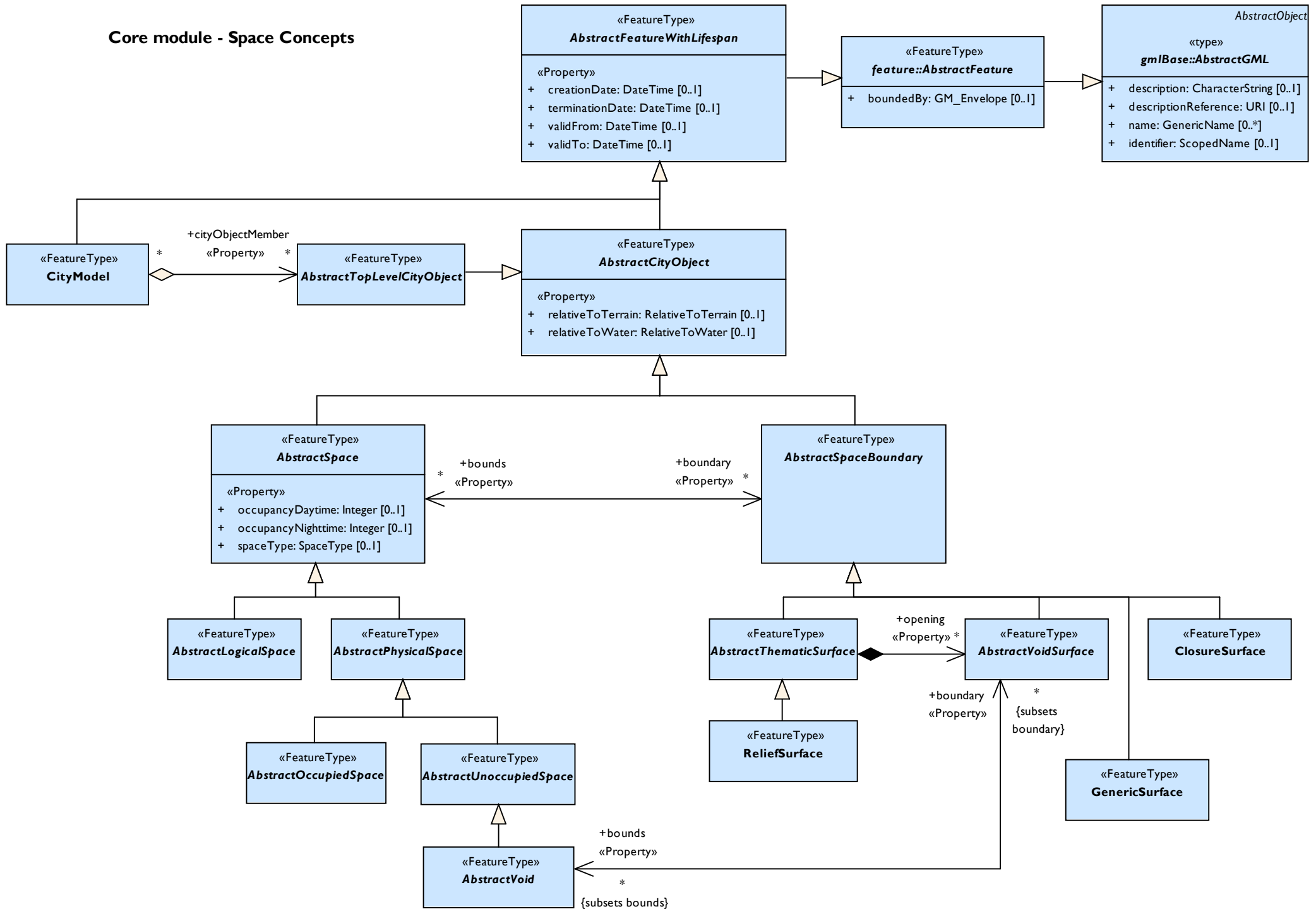
Usage of ISO and OGC standards in CityGML



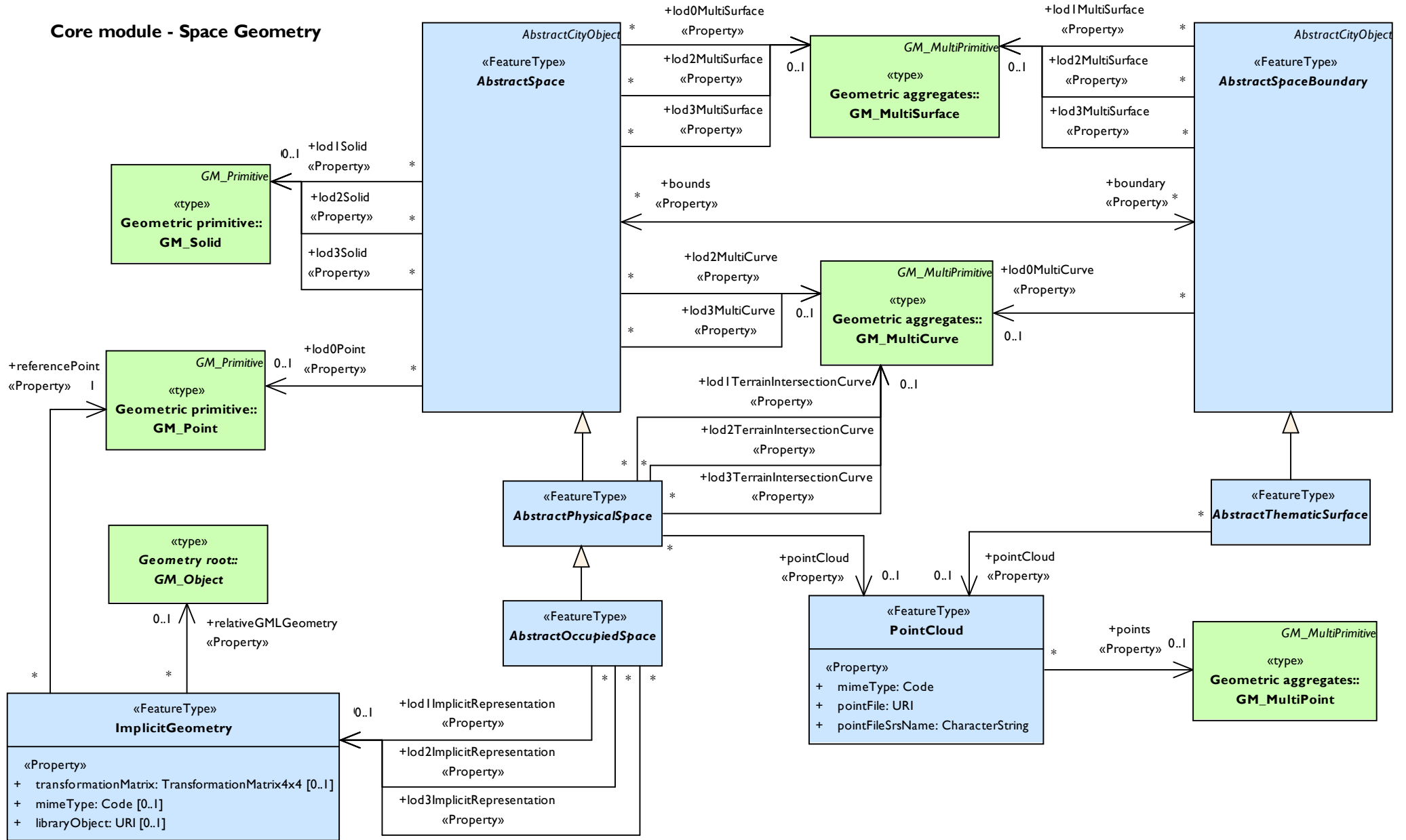
Core module - Overview



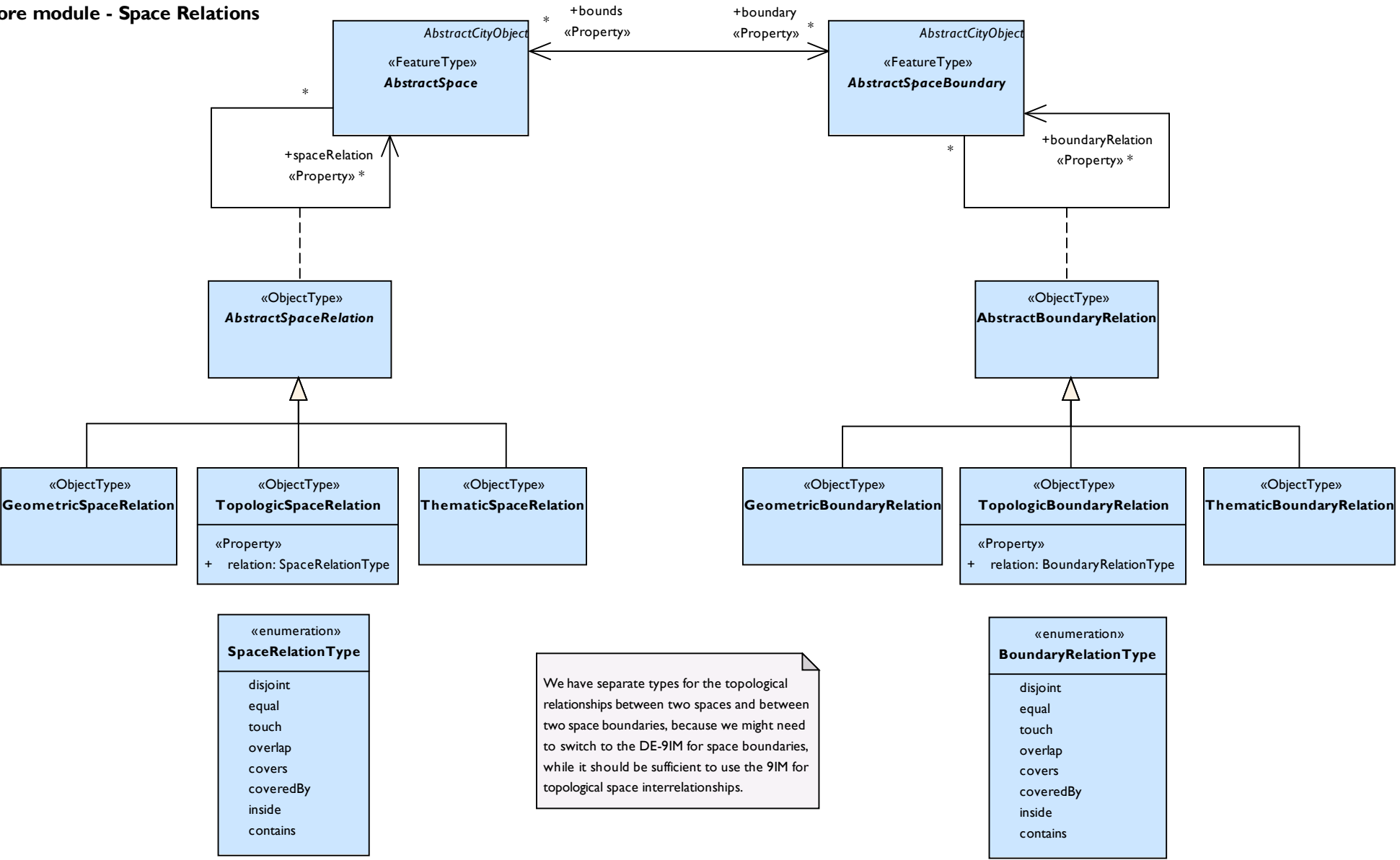
Core module - Space Concepts



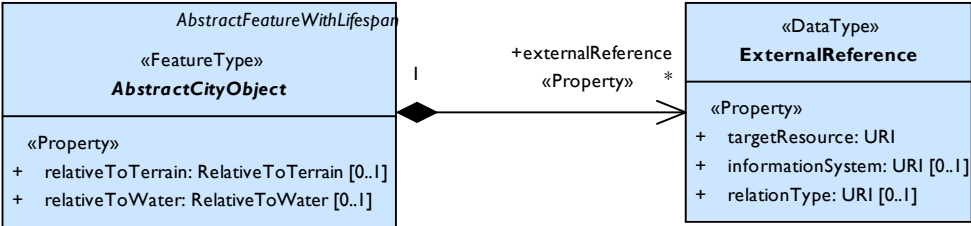
Core module - Space Geometry



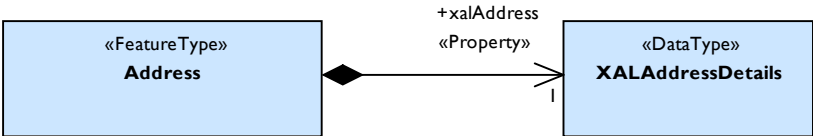
Core module - Space Relations



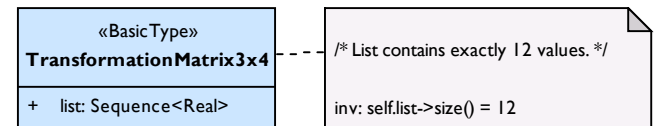
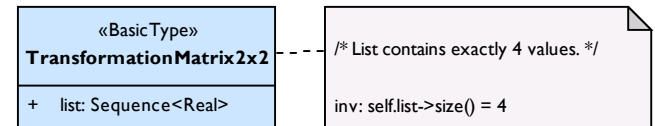
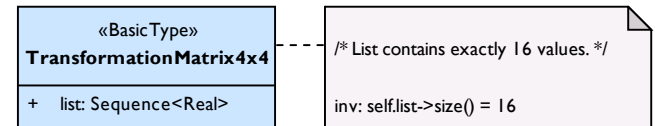
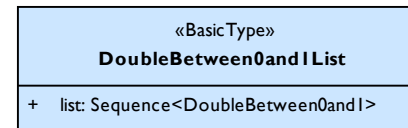
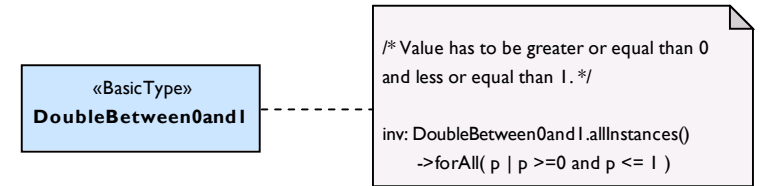
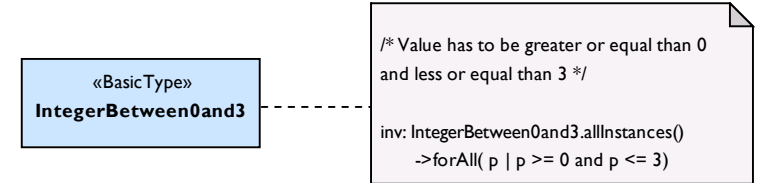
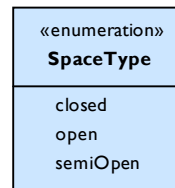
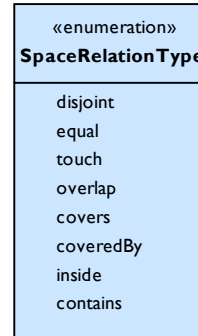
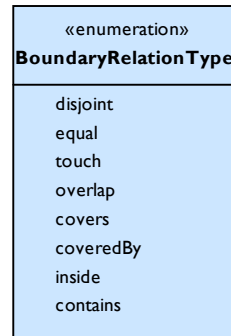
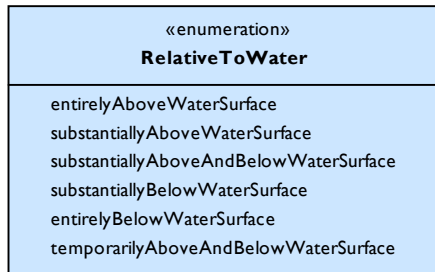
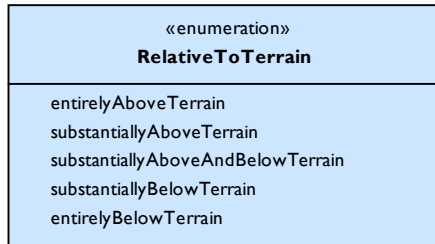
Core module - Miscellaneous

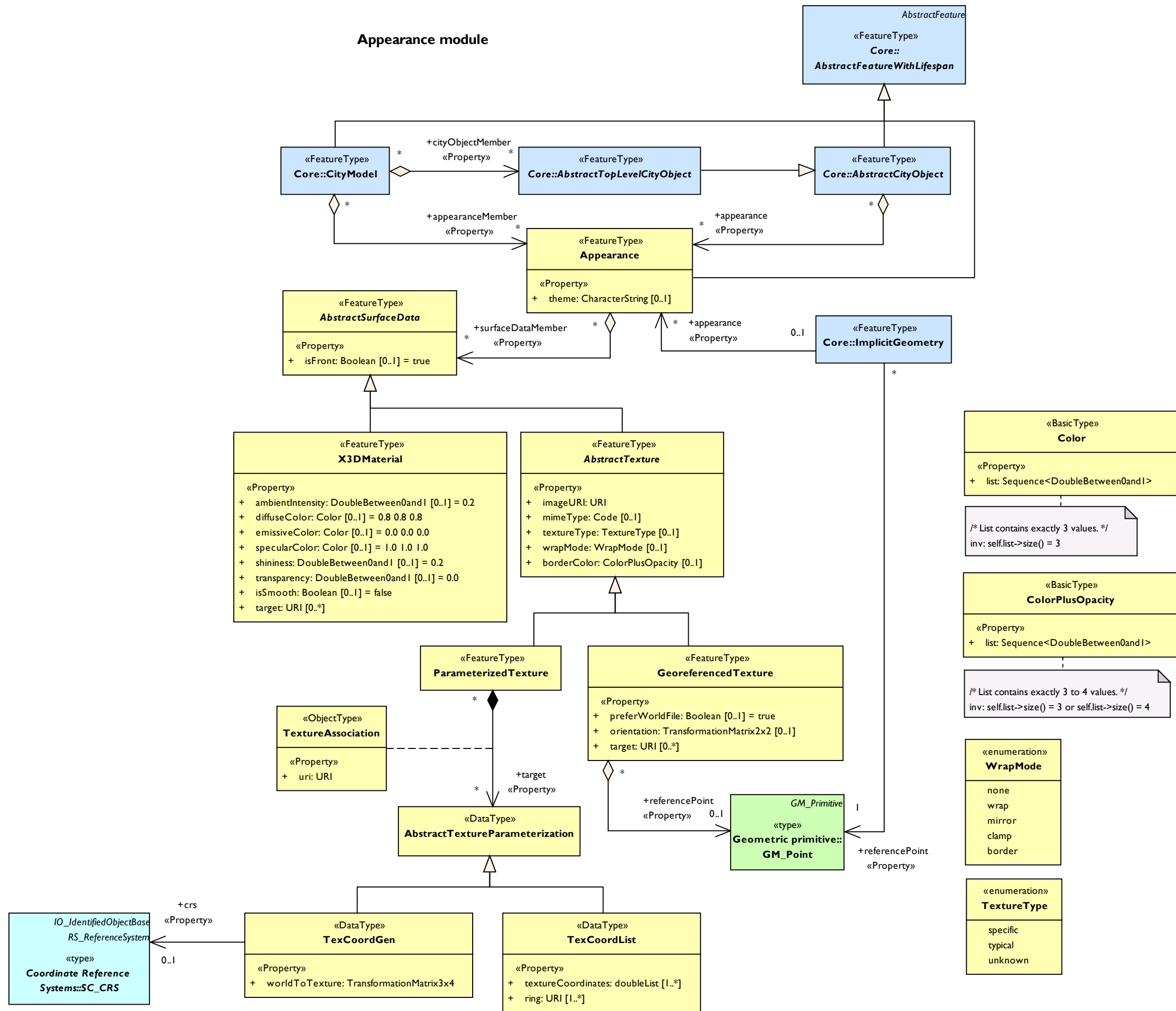


ExternalReference is now extended by an optional **relationType** which can link to some external definition of the type of relation (e.g. the *sameAs* relation from OWL). Hence, **ExternalReferences** can now be used to express relations similar to RDF.



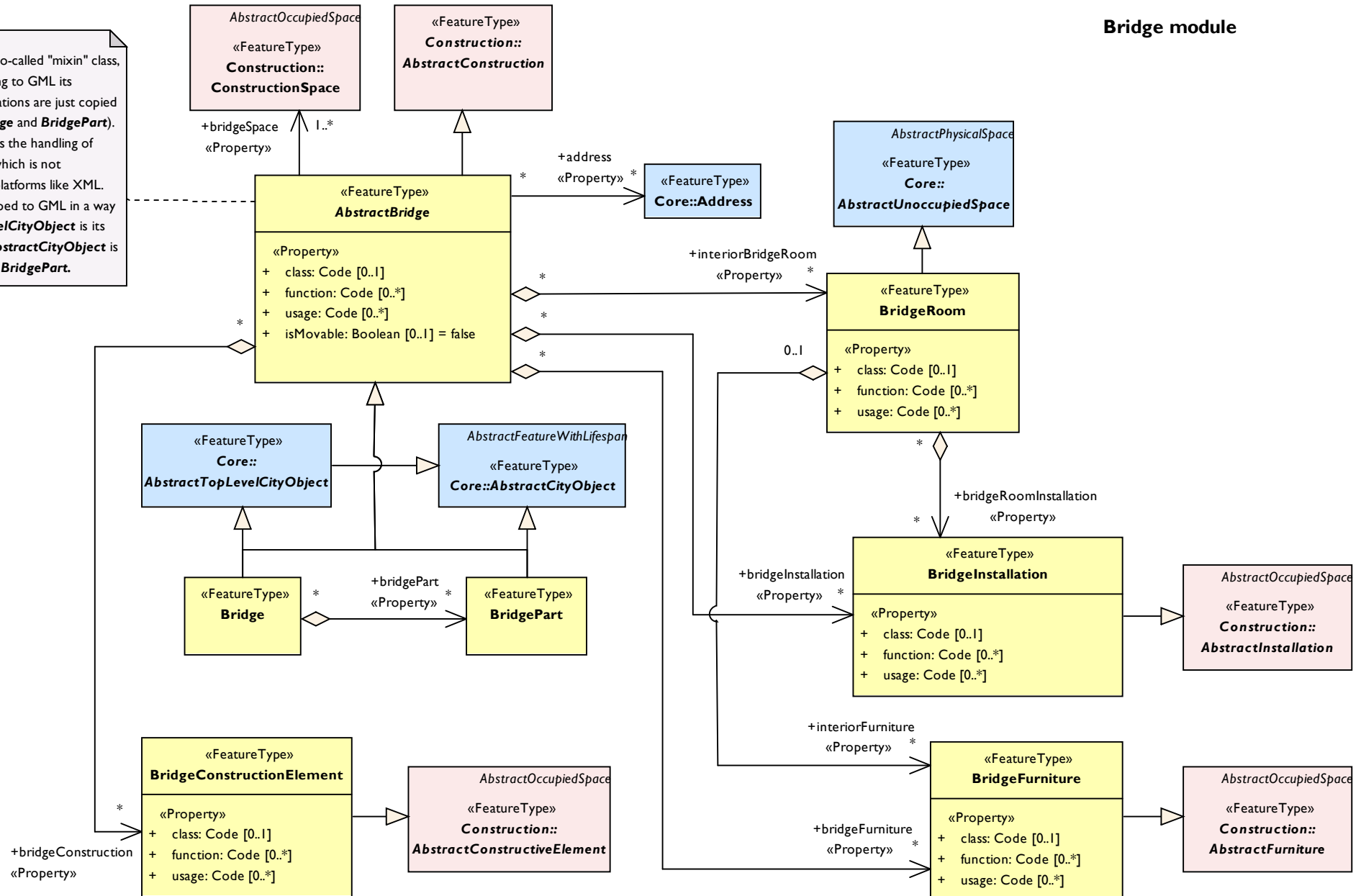
Core module - Basic Types and Enumerations

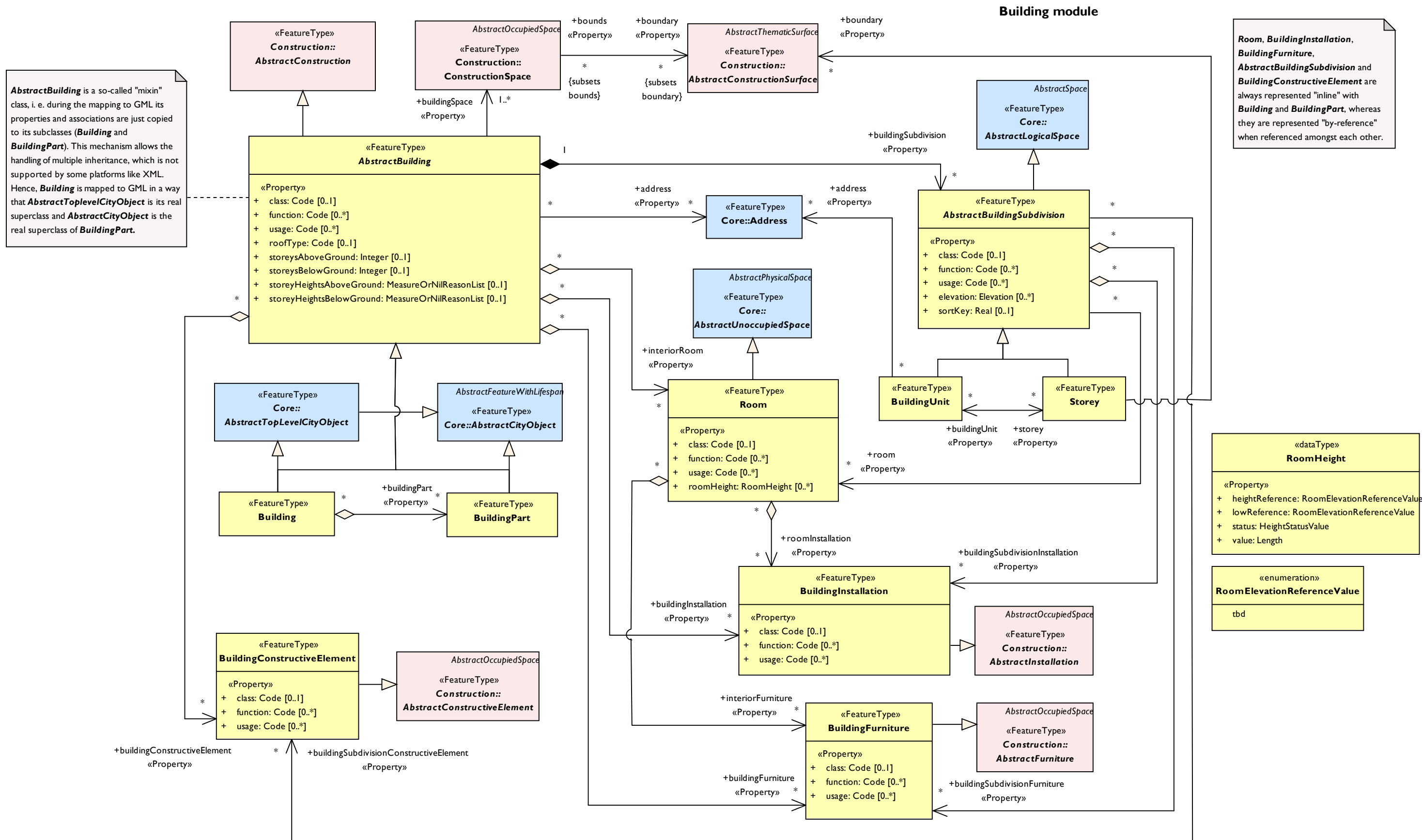




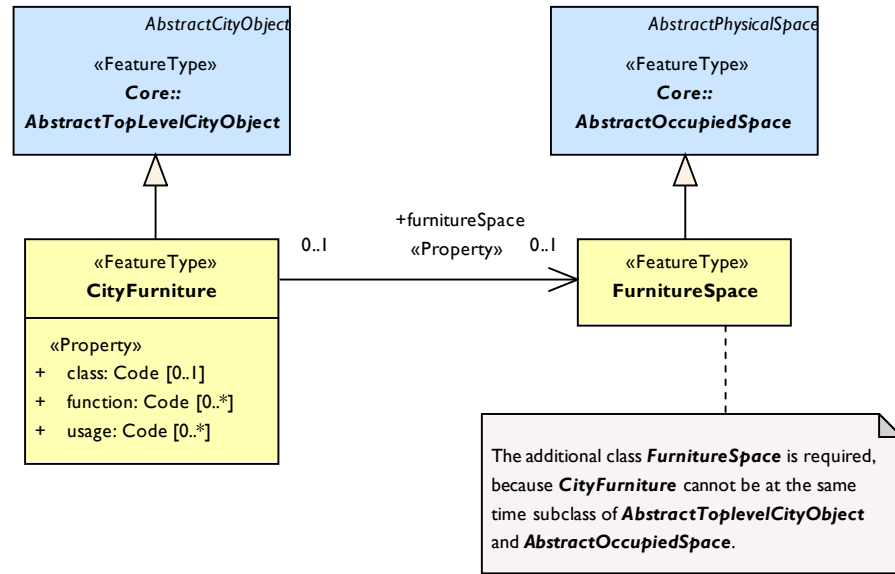
Bridge module

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

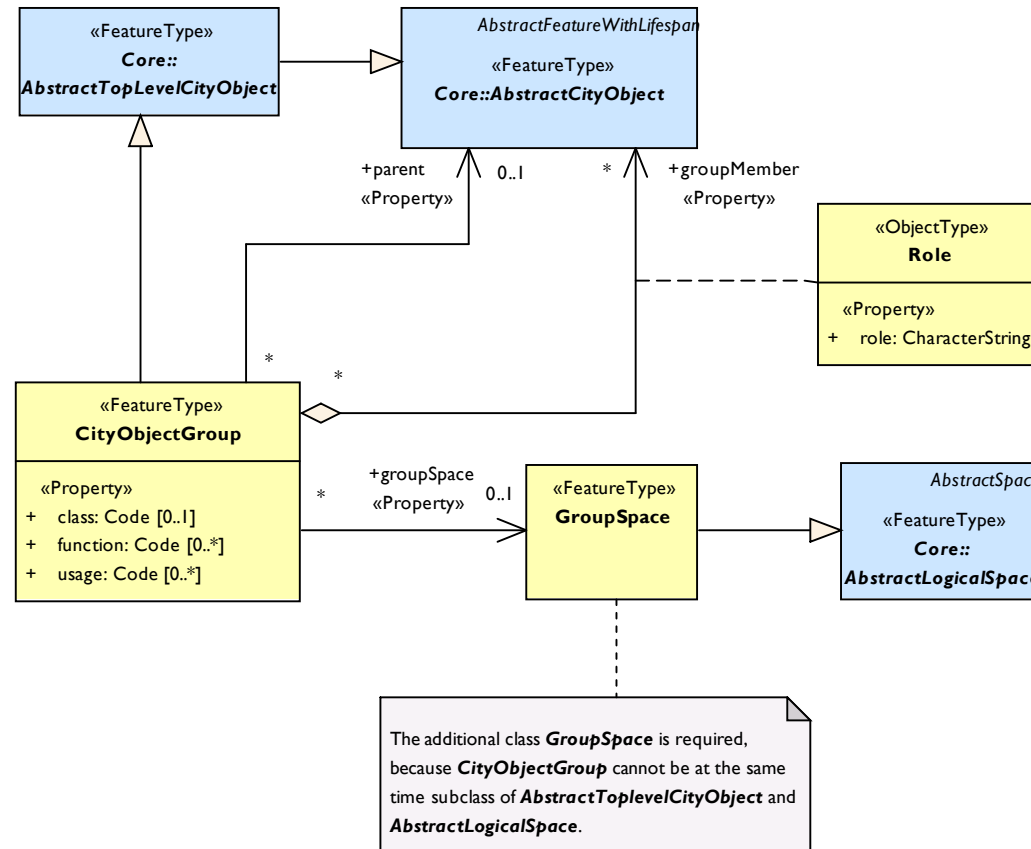




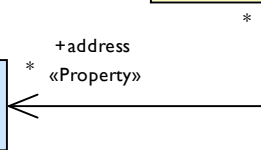
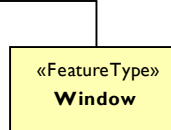
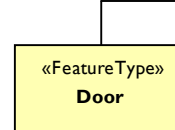
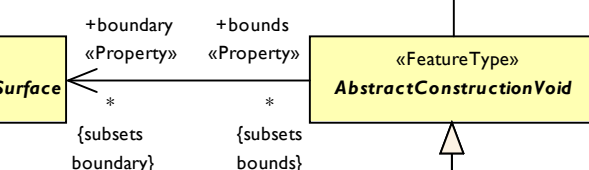
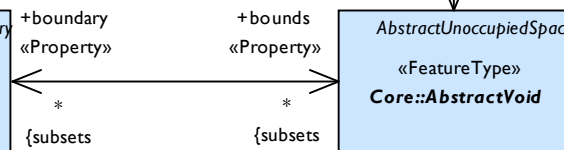
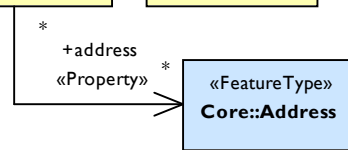
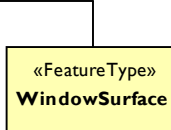
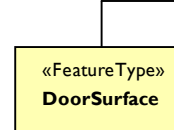
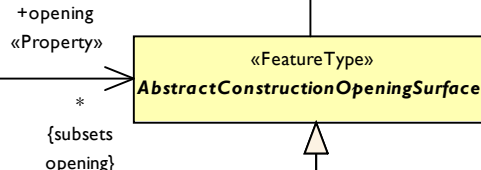
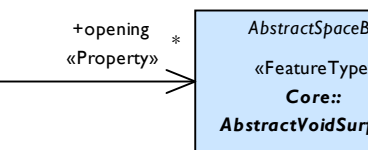
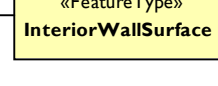
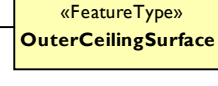
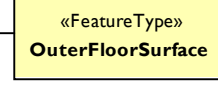
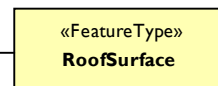
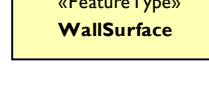
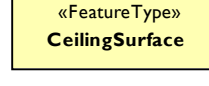
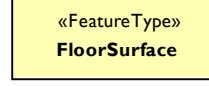
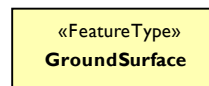
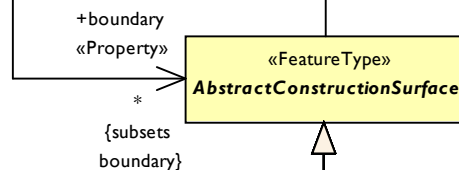
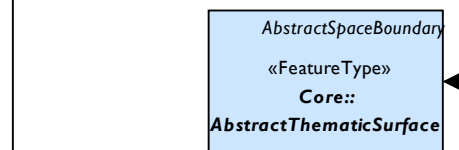
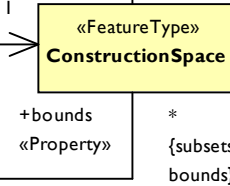
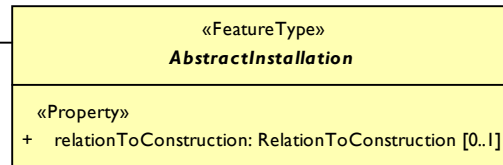
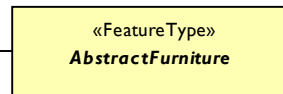
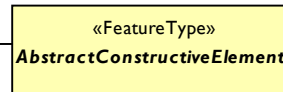
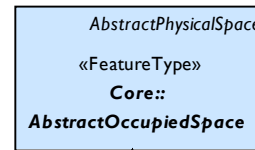
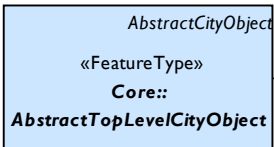
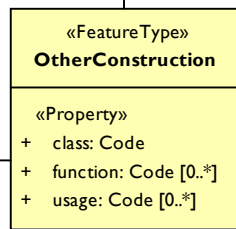
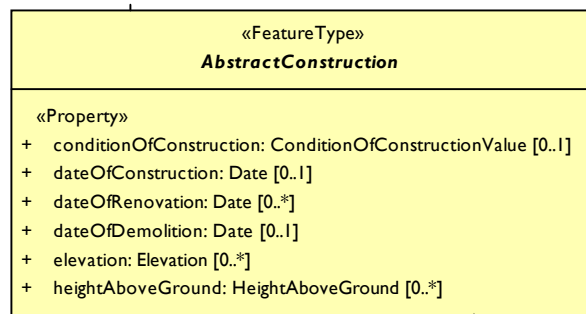
CityFurniture module



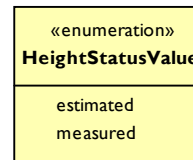
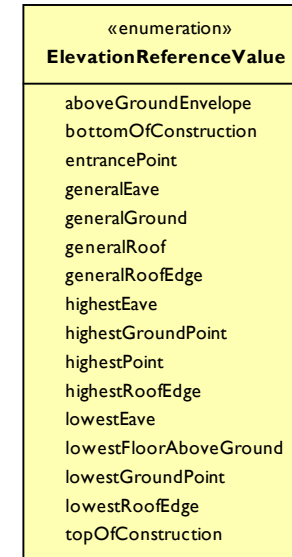
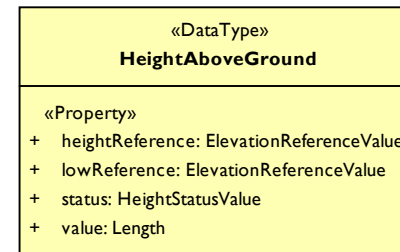
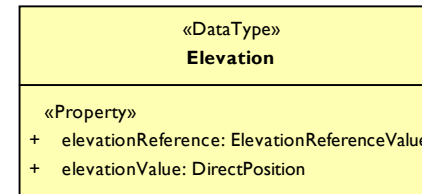
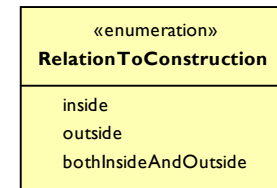
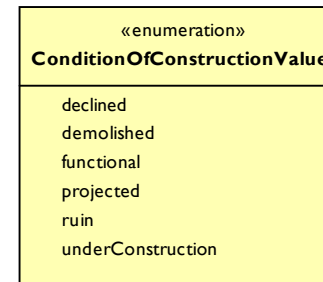
CityObjectGroup module



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.

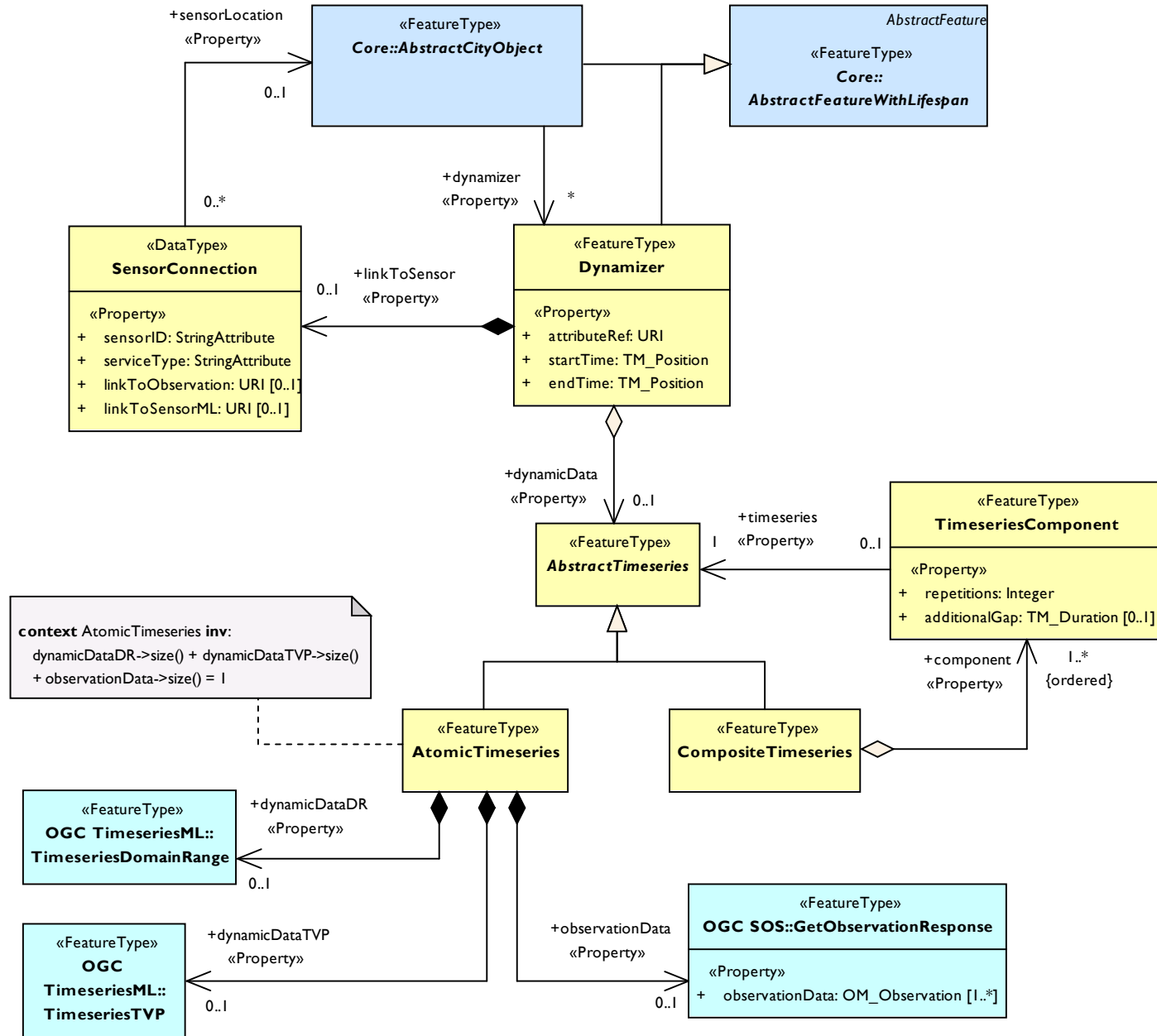


+opening
«Property»
*



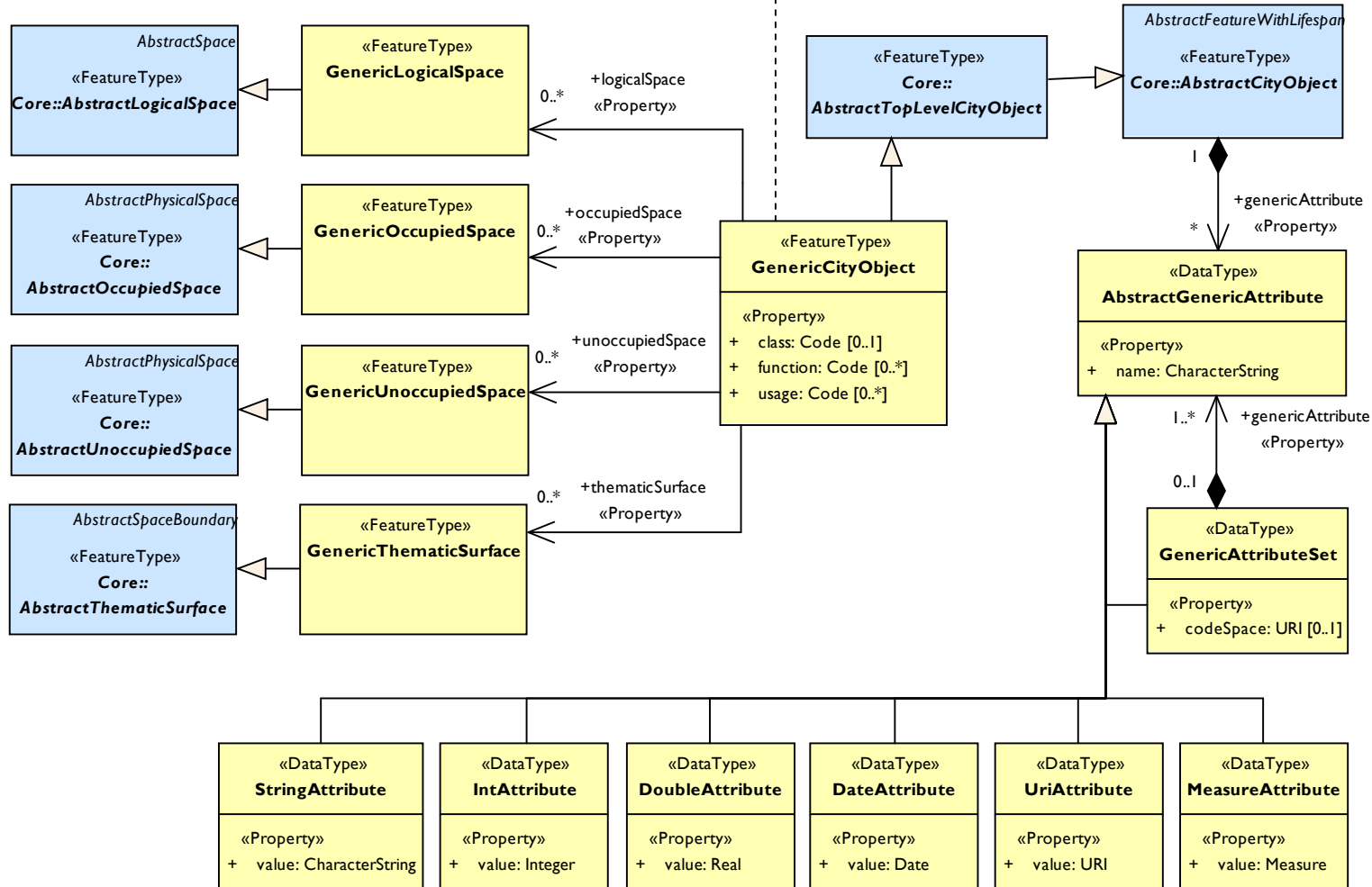
It needs to be discussed further within the SWG, whether a **Door** is a **Void** or an **Opening**,

Dynamizer module

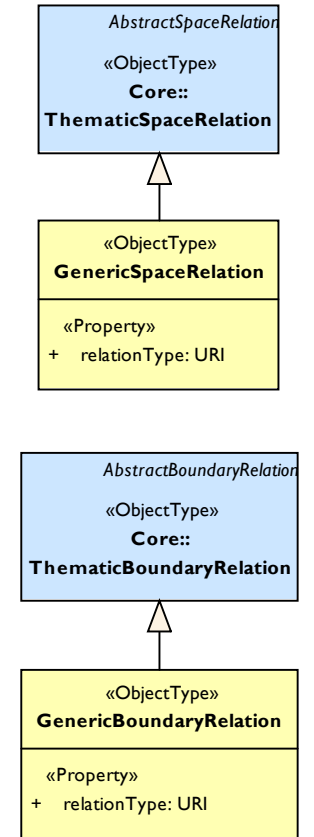


Generics module

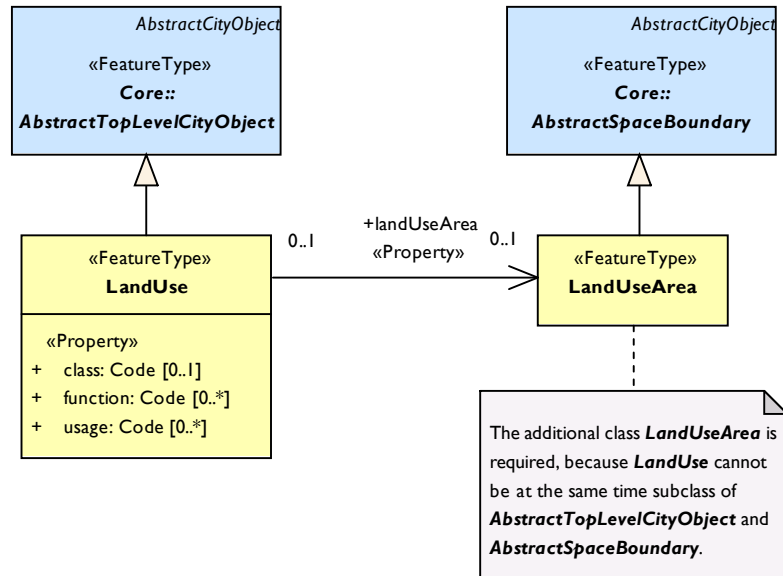
An instance of **GenericCityObject** can only be associated either with one or several instances of one of the Space classes or with one or several instances of **GenericThematicSurface**.

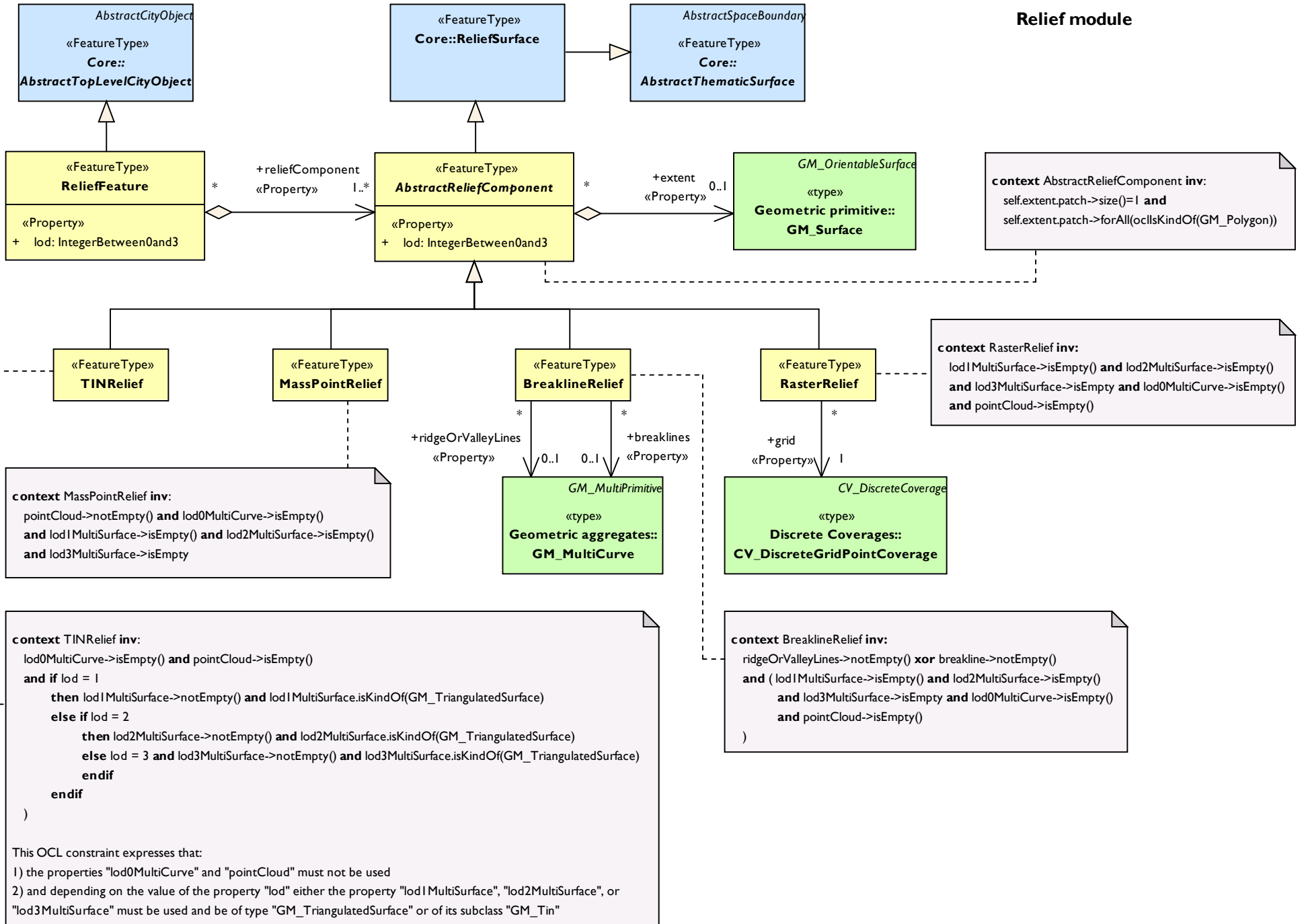


The **GenericSpaceRelation** and **GenericBoundaryRelation** can be used to represent an arbitrary relationship between two spaces or two space boundaries respectively. The relation type is expressed by an URI. Such relations could directly be mapped to RDF triples, where the source is pointing to the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

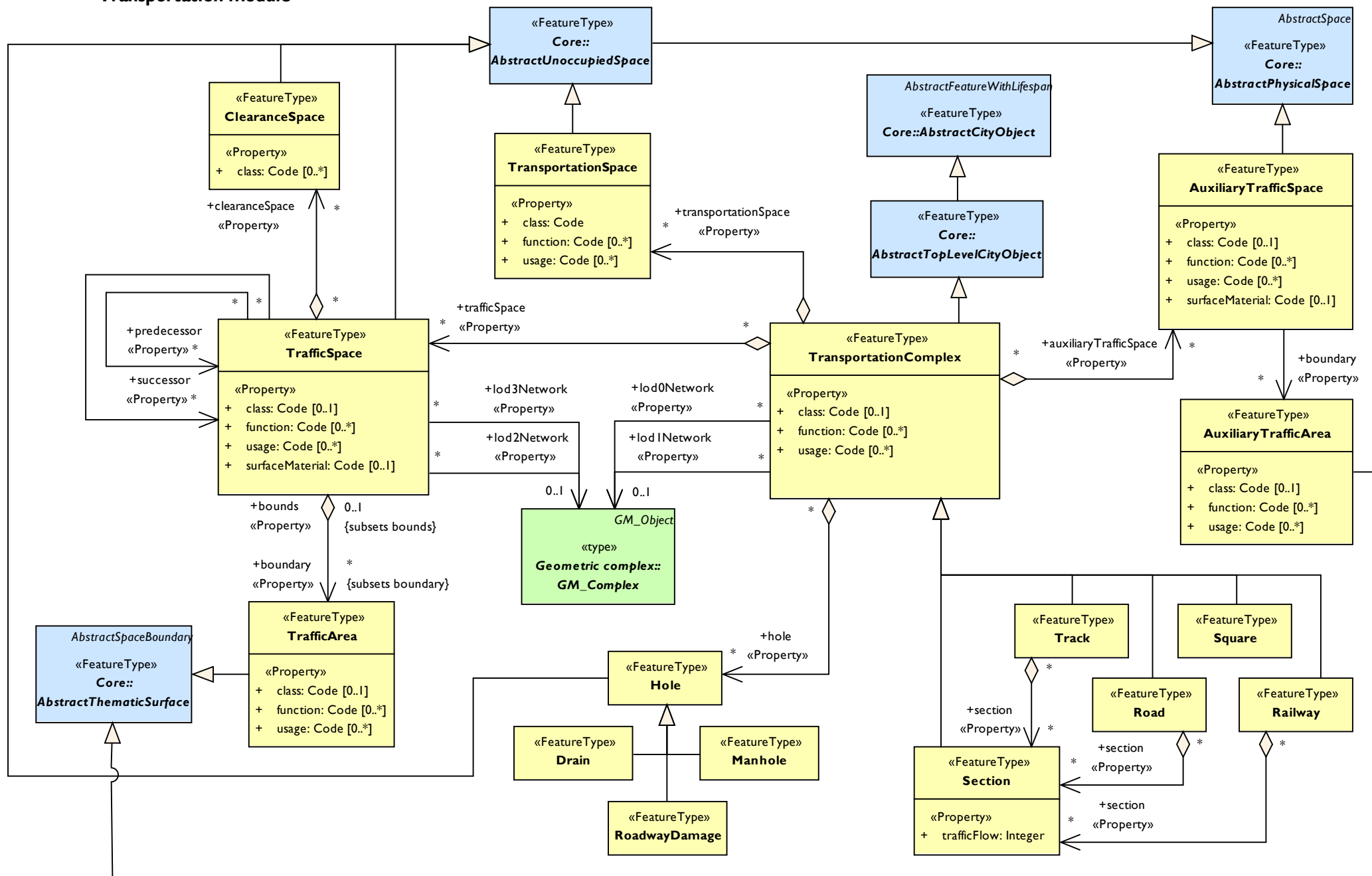


LandUse module



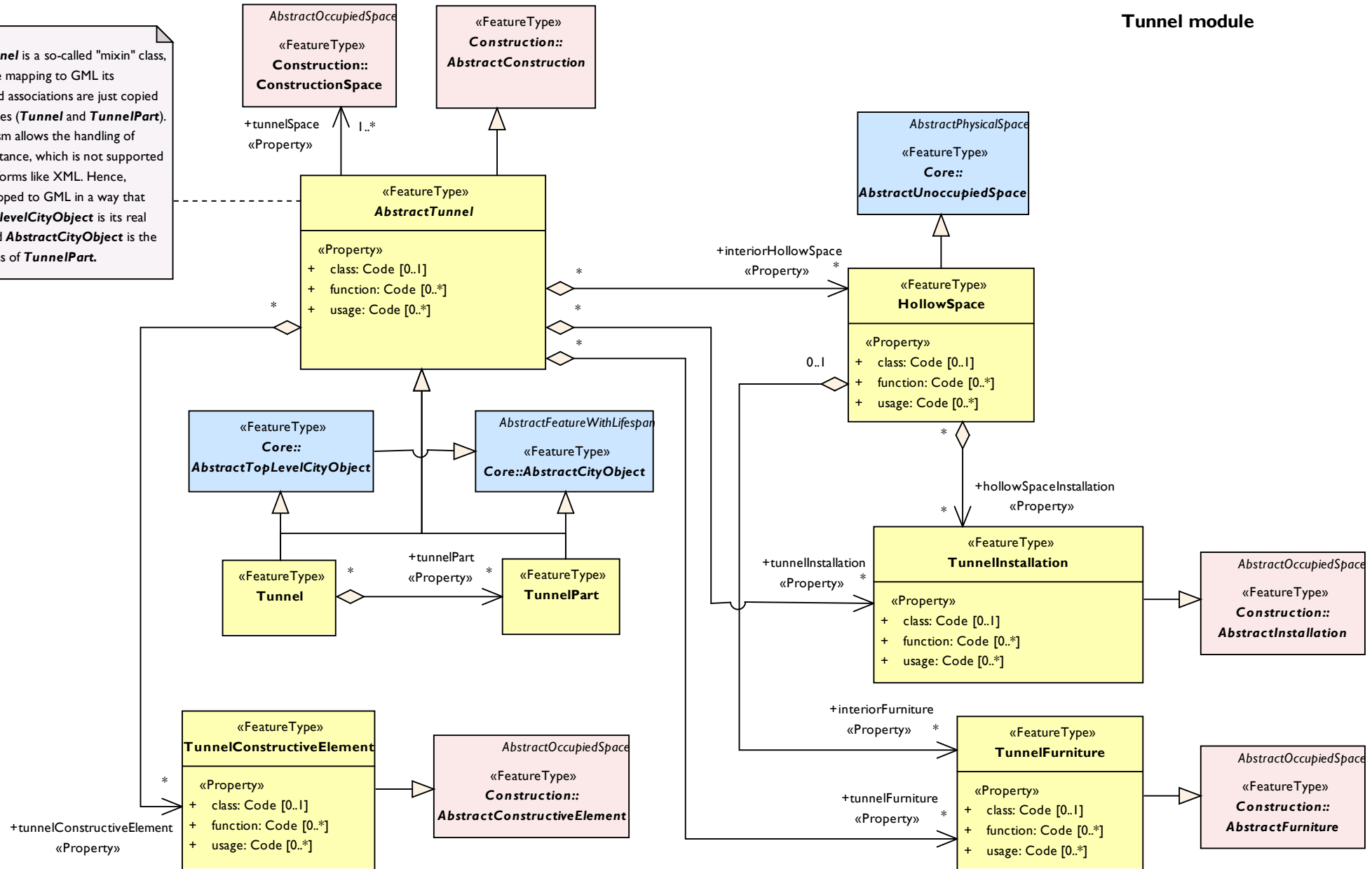


Transportation module

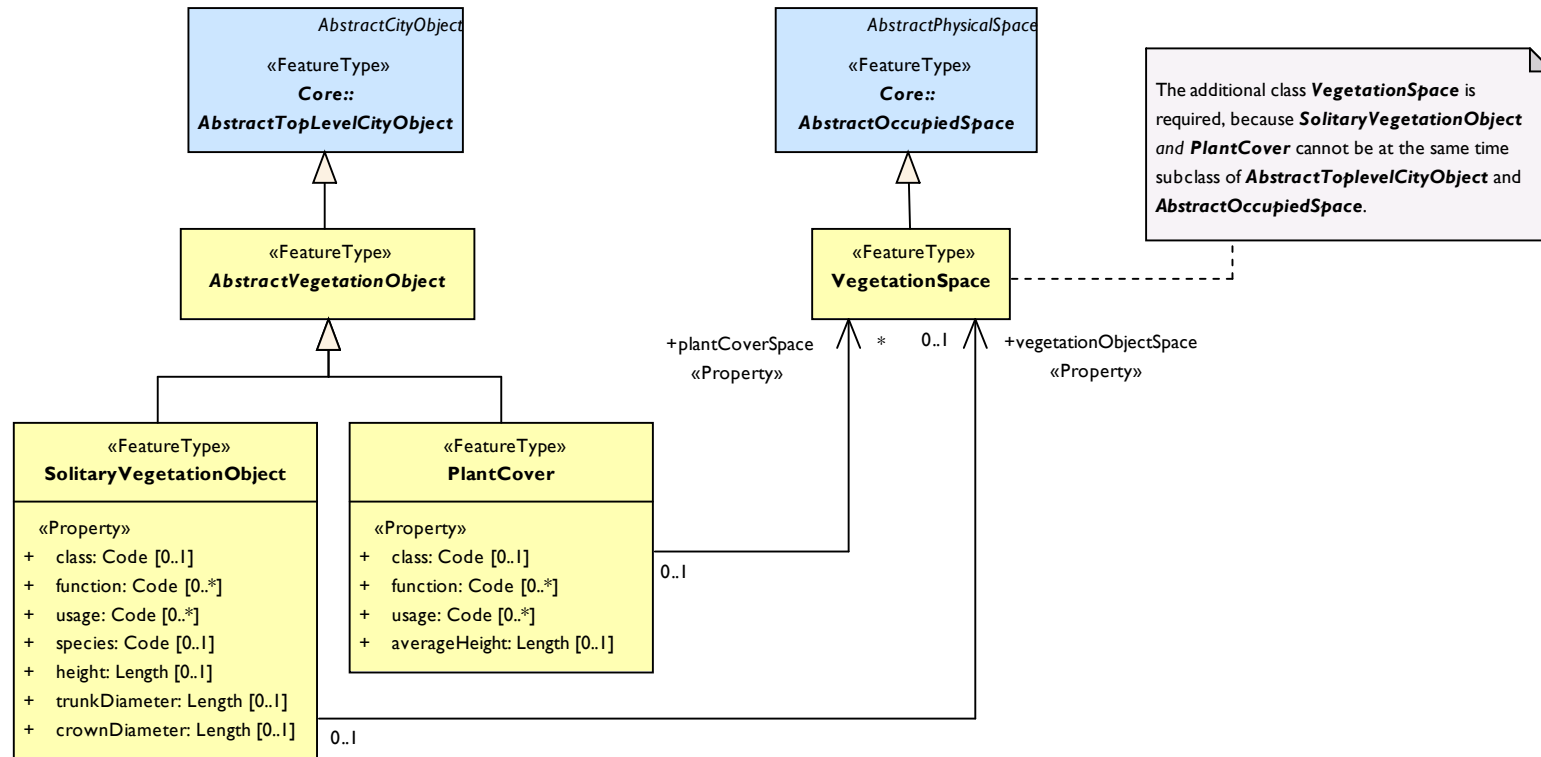


Tunnel module

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

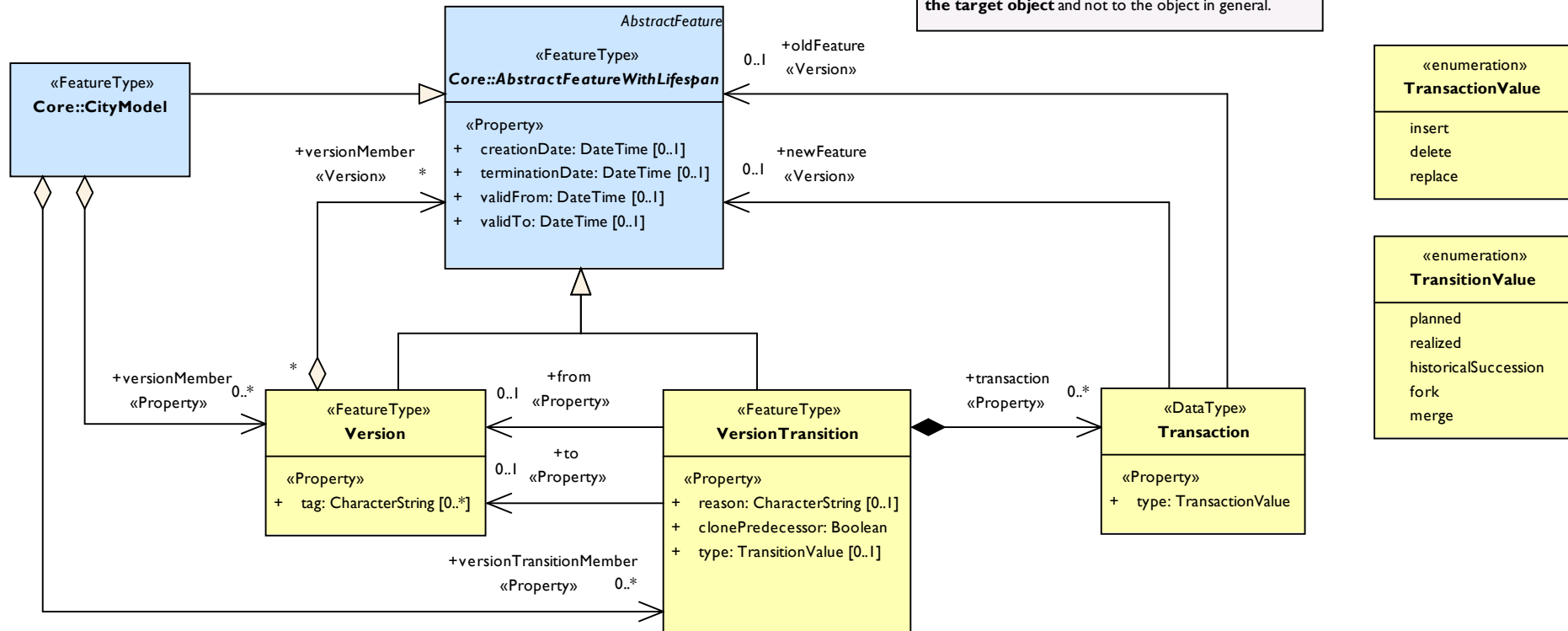


Vegetation module



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

