Introduction to CityGML and the new CityGML 3.0 UML model

Tatjana Kutzner

Chair of Geoinformatics
Prof. Thomas H. Kolbe
Technical University of Munich

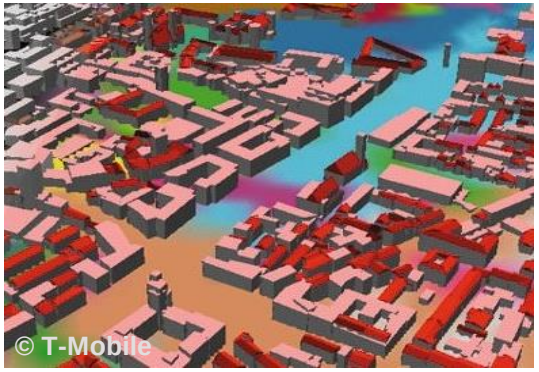
kutzner@tum.de

OGC CityGML Hackathon, 11/12 June 2019, London

Applications of Virtual 3D City Models

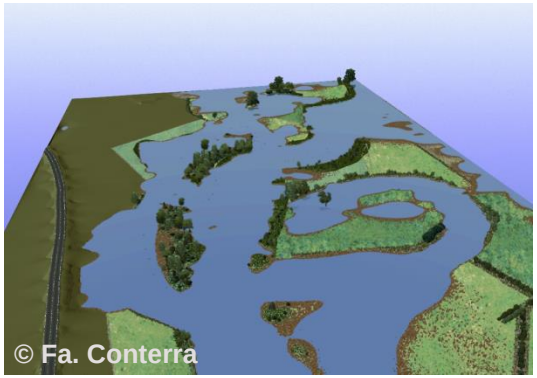
Urban planning

Cellular network planning

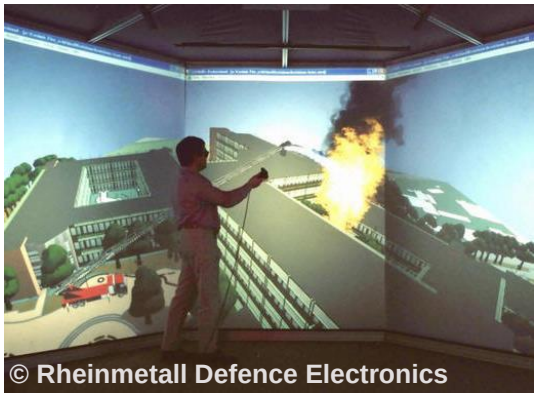


3D navigation

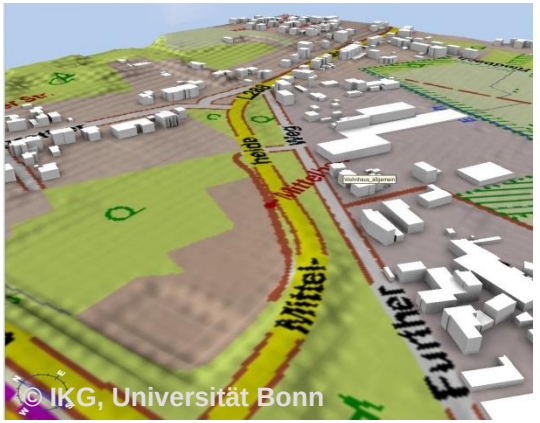
Disaster management



Training simulators



Noise pollution mapping



City Geography Markup Language – CityGML

Application independent Geospatial Information Model

for semantic 3D city and landscape models

- comprises **different thematic areas** (buildings, vegetation, water, terrain, traffic etc.)
- **data model** (UML) + **Exchange format** (based on GML 3)

CityGML represents

- 3D geometry, 3D topology, semantics and appearance
- in several discrete scales (Levels of Detail, LOD)

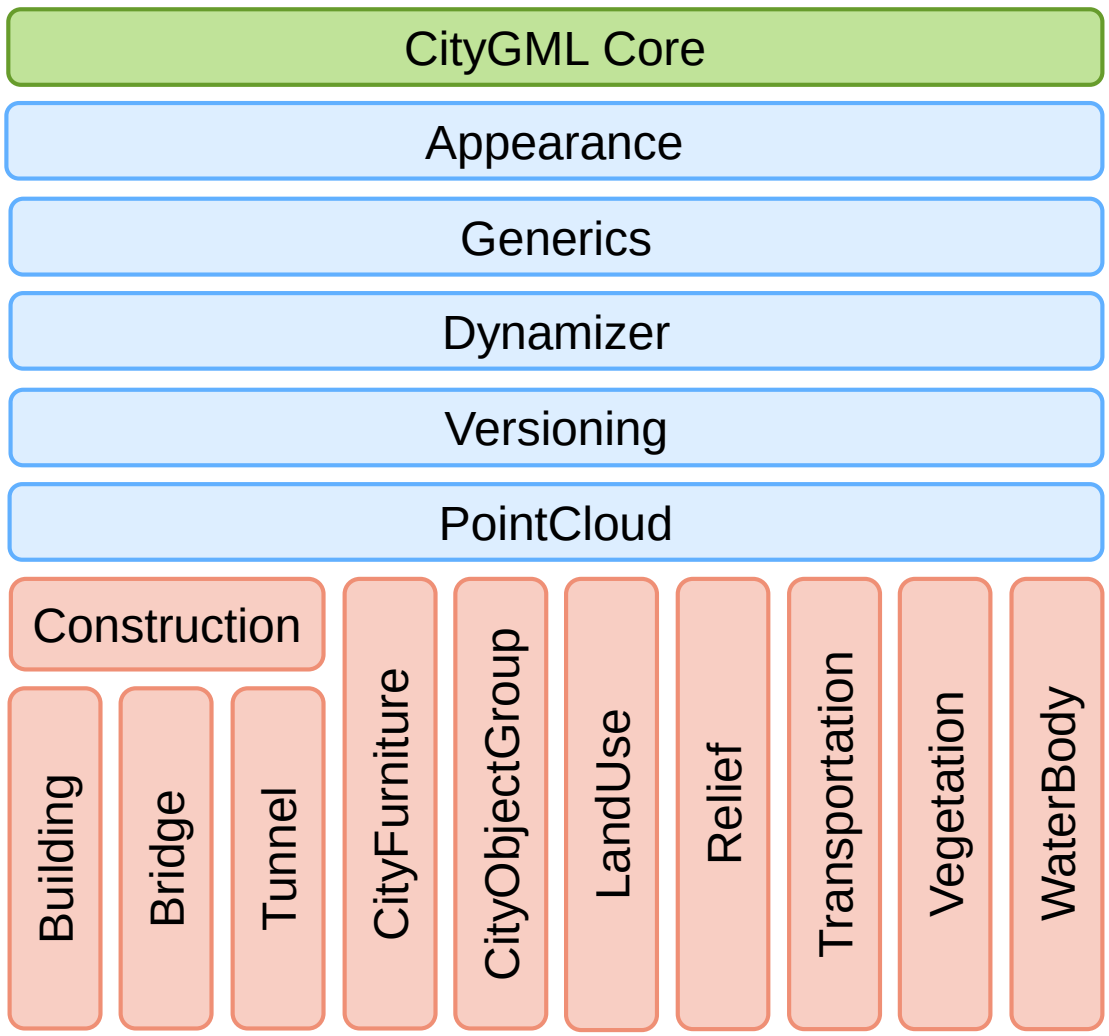
CityGML is an international standard of the OGC

- Version 1.0.0 was adopted in 2008, version 2.0.0 was adopted in 2012
- **Version 3.0.0 is currently under development and will be released end of 2019**

(Inter)national Usage / Availability of CityGML

- **Cities / Municipalities**
  - e.g. almost all German cities with 3D city models; Rotterdam, Zürich, Geneva, Paris, Marseille, Istanbul, Vancouver, Montreal, Kuala Lumpur, Yokohama, Doha, New York
- **Organisations**
  - e.g. IGN France, Ordnance Survey UK, State Mapping Agencies of Bavaria, Baden-Wuerttemberg, Hesse, Rhineland-Palatinate, North Rhine-Westphalia, BIMTAŞ in Istanbul, **many companies, research institutes, and universities**
- CityGML is **reference model** in the European **INSPIRE** initiative (→ full EU coverage)
  - INSPIRE Building model is based on CityGML
- The official national and municipal 3D geoinformation standards of Germany and the Netherlands base on CityGML

CityGML 3.0 Module Overview



Defines base concepts and geometries inherited by all other modules

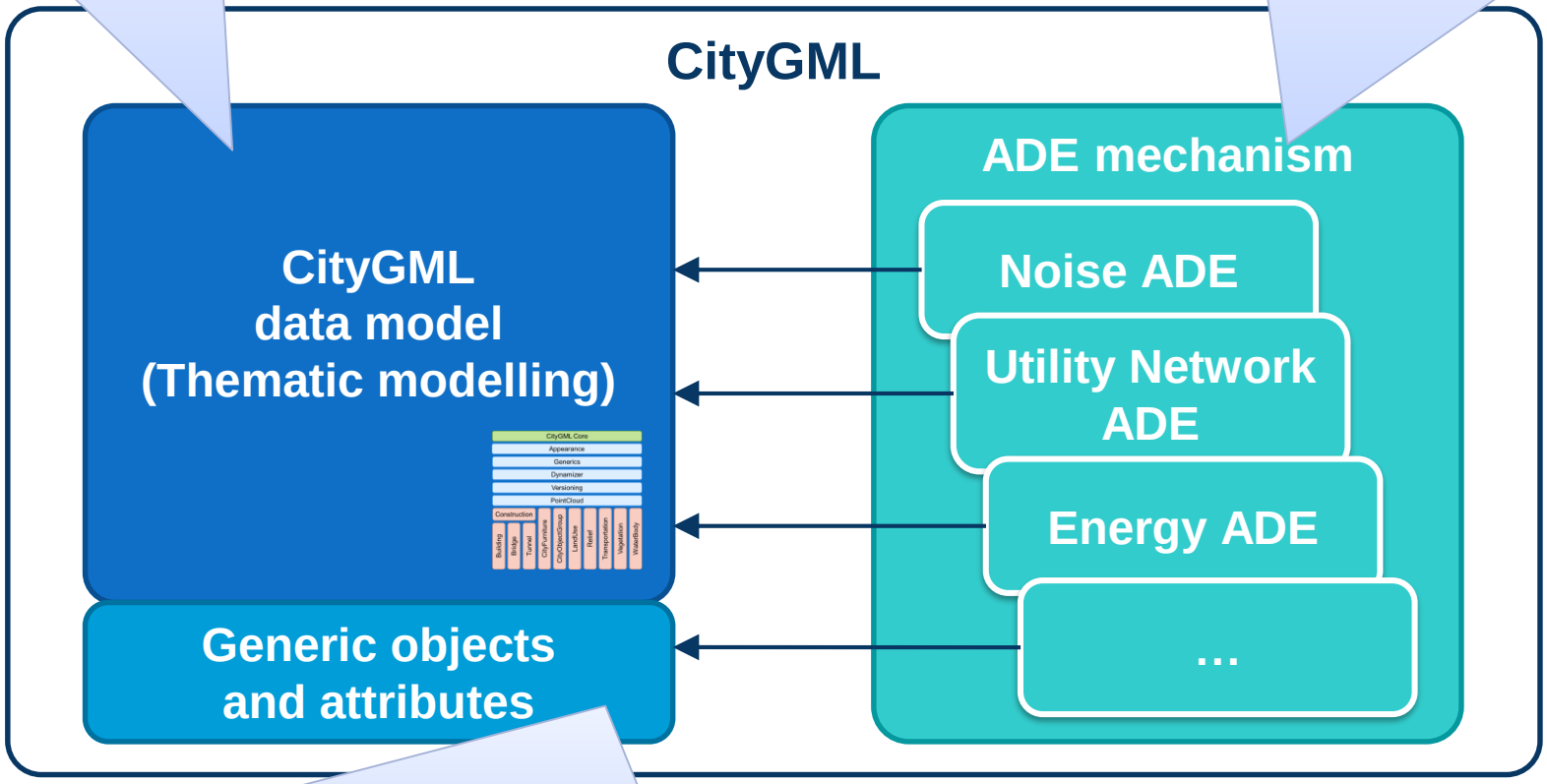
Define concepts applicable to all thematic modules

Thematic modules

CityGML Extension Mechanisms

Structured, strict data model consisting of **well-defined classes** and code lists

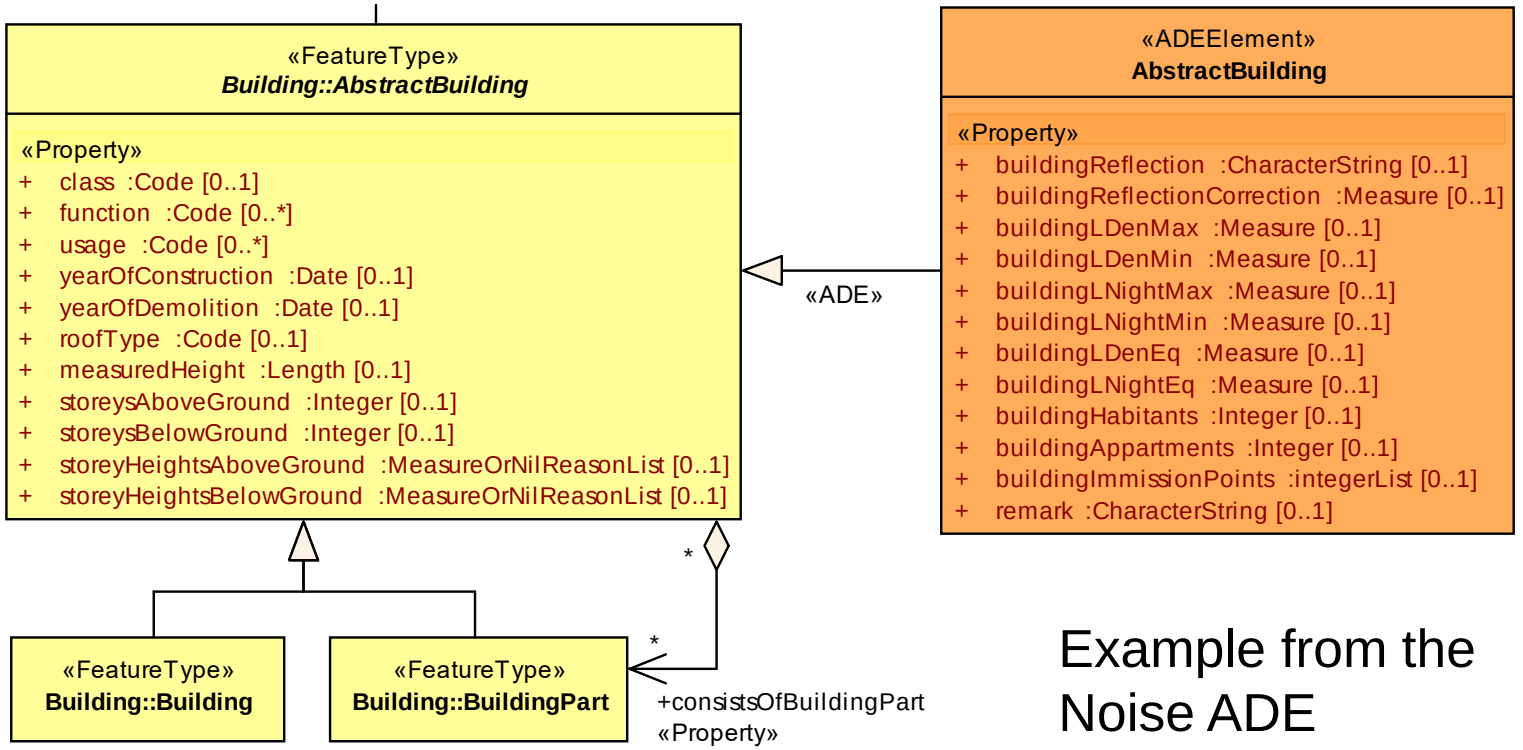
Systematic extension mechanism, allows for extending every CityGML object type by additional attributes and for introducing new object types



Semi-structured extension mechanism, allows for flexibly extending the data model without making a schema modification necessary (“Extension during run-time”)

How the ADE mechanism works (I)

- Within a separate ADE model the new attributes are modelled as subclasses of existing classes
  - Subclasses receive the stereotype «ADEElement»
  - Generalisation relationships receive the stereotype «ADE»



Example from the Noise ADE

How the ADE mechanism works (II)

- During encoding the new attributes are injected into the respective superclass → superclass strategy

```
<cityObjectMember>
 <bldg:Building gml:id="ef6e19e3-c412-440b-8ba9">
  <bldg:function>1060</bldg:function>
  <bldg:measuredHeight uom="m">2.38</bldg:measuredHeight>
  . . .
  <noise:buildingReflection>Fassade</noise:buildingReflection>
  <noise:buildingReflectionCorrection uom="dB">
    3.23
  </noise:buildingReflectionCorrection>
  <noise:buildingLDenMax uom="dB">10</noise:buildingLDenMax>
  <noise:buildingLDenMin uom="dB">30</noise:buildingLDenMin>
  <noise:buildingLDenEq uom="dB">20</noise:buildingLDenEq>
  . . .
 </bldg:Building>
</cityObjectMember>
```

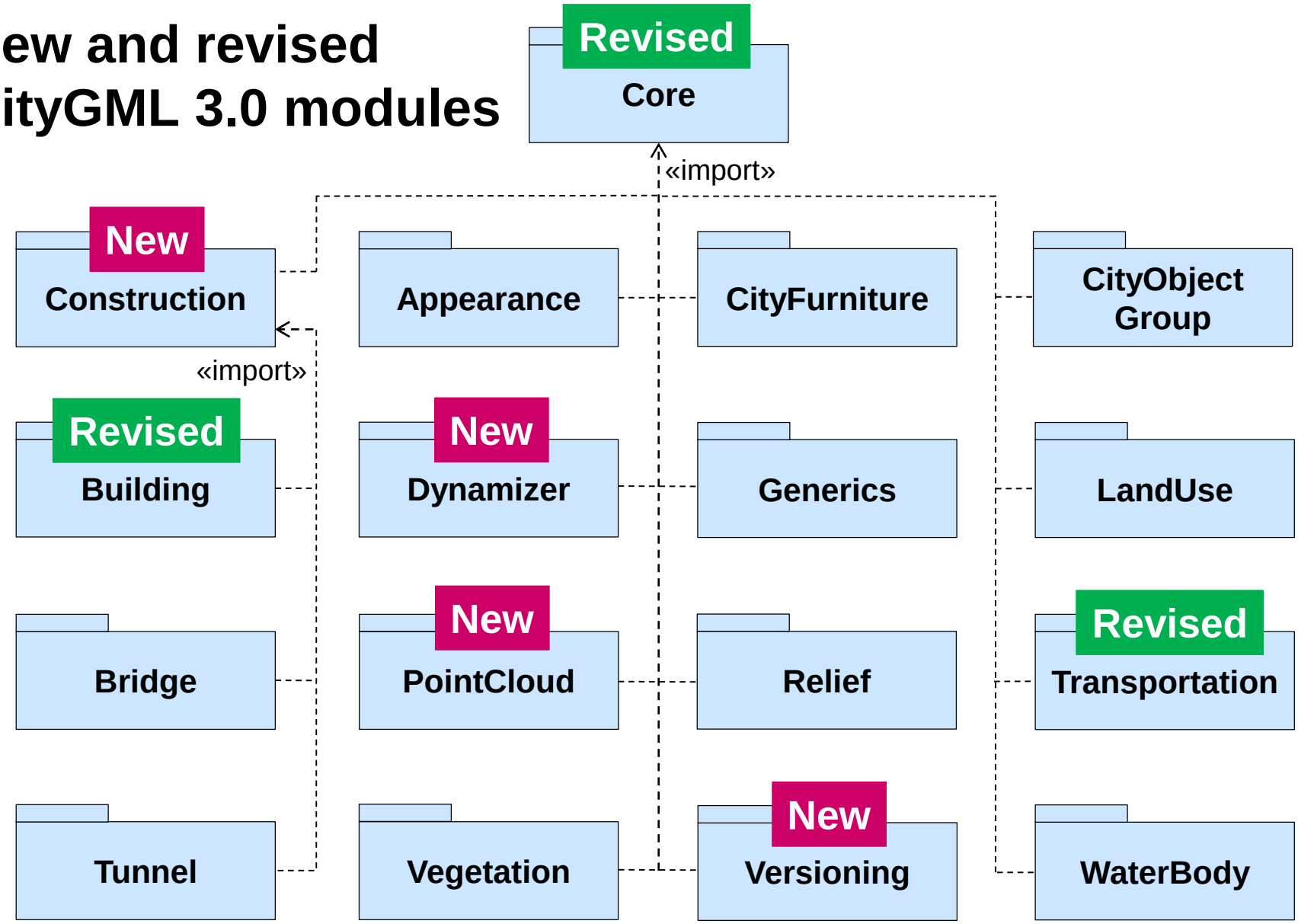
standard CityGML attributes of a building

Noise ADE attributes

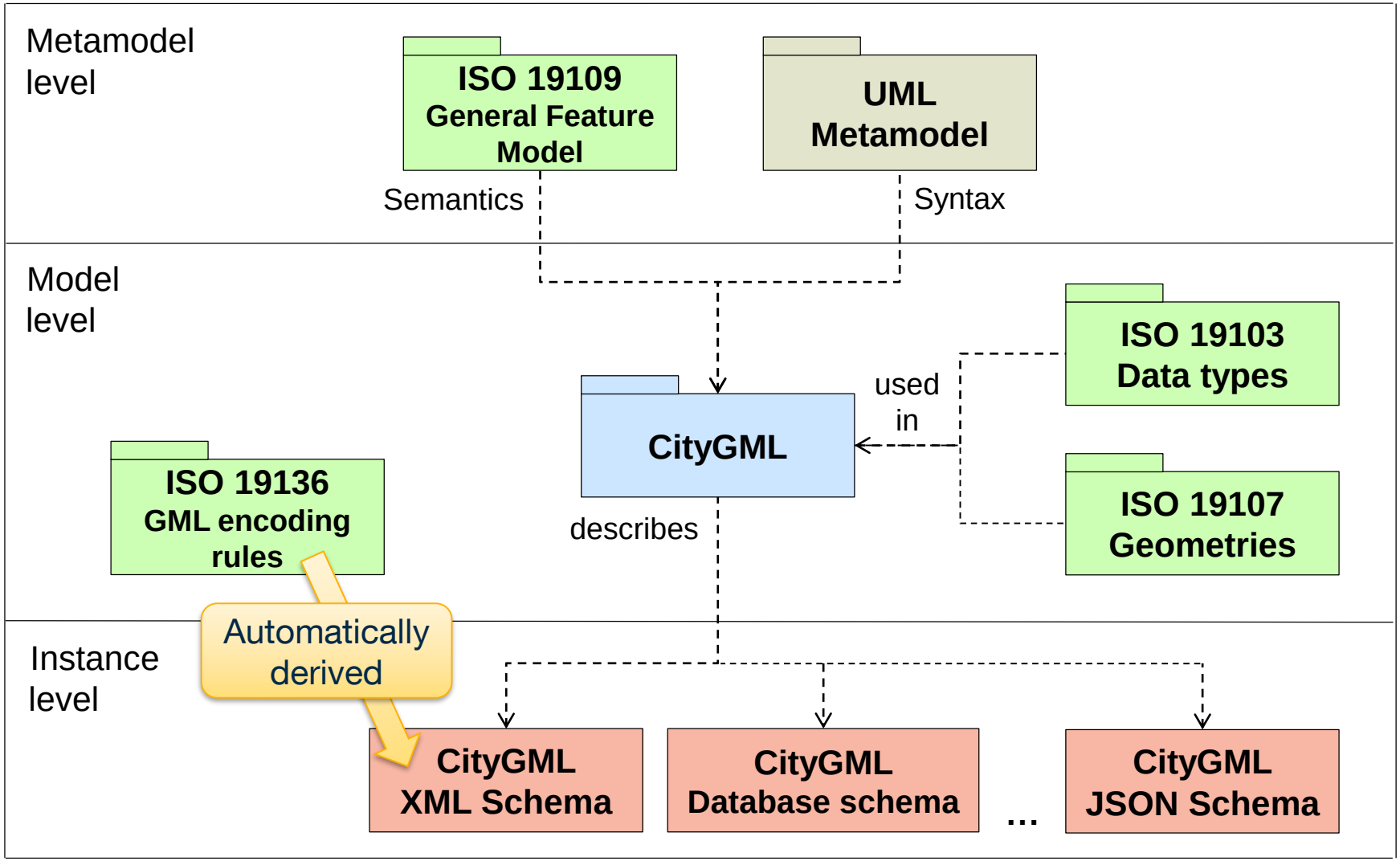
Characteristics of CityGML 3.0

- **New and revised modules**: Improved support for using 3D city models in urban planning, simulations and analyses
- **Less redundancy**: Concepts used in several modules are integrated and provided centrally via inheritance
- All city objects are based on **two new central concepts**: Spaces and SpaceBoundaries (represented by the classes AbstractSpace and AbstractThematicSurface)
- **Better interoperability** with other standards (IndoorGML, IFC, RDF, LADM, INSPIRE)
- **Model-driven approach**: ISO-compliant UML model + automatic derivation of exchange formats
  - At least two specifications: - CityGML 3.0 Conceptual Model specification
    - CityGML 3.0 GML Encoding specification
- **Backwards compatibility** with CityGML 1.0 and 2.0

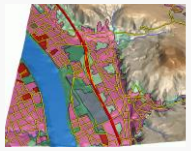
New and revised CityGML 3.0 modules



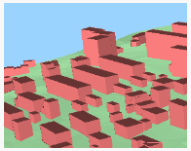
ISO-compliance of CityGML 3.0



New LOD concept: 4 levels of details



LOD0 – Regional, landscape model **+ interior**
- 2.5D Digital terrain model



LOD1 – City, regional model
- Prismatic buildings without roof structures **+ interior**

LOD2 – City districts, site model
- Simple buildings with detailed roof structures **+ interior**

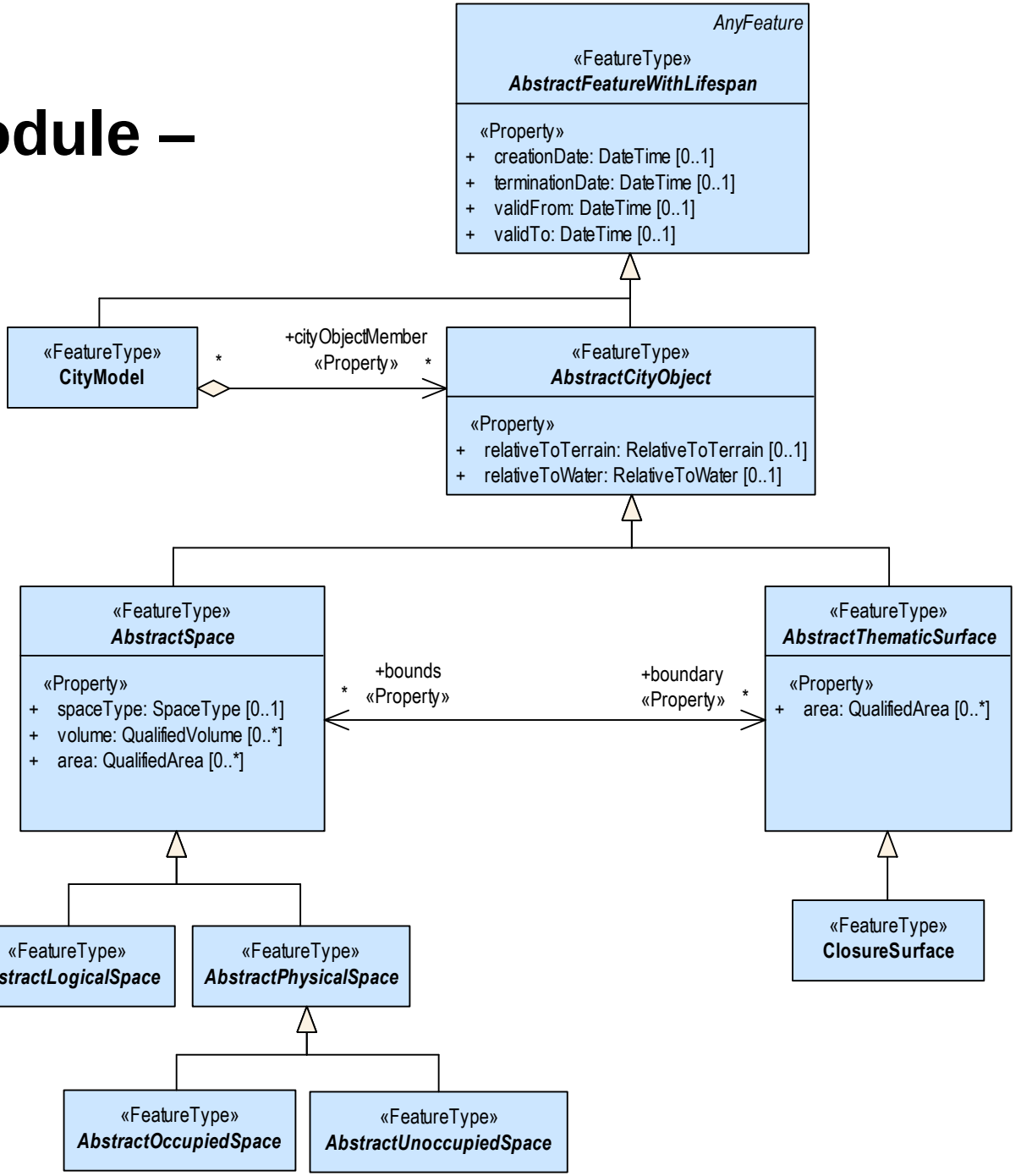
LOD3 – Architectural models (exterior) **+ interior**
- Detailed architectural models

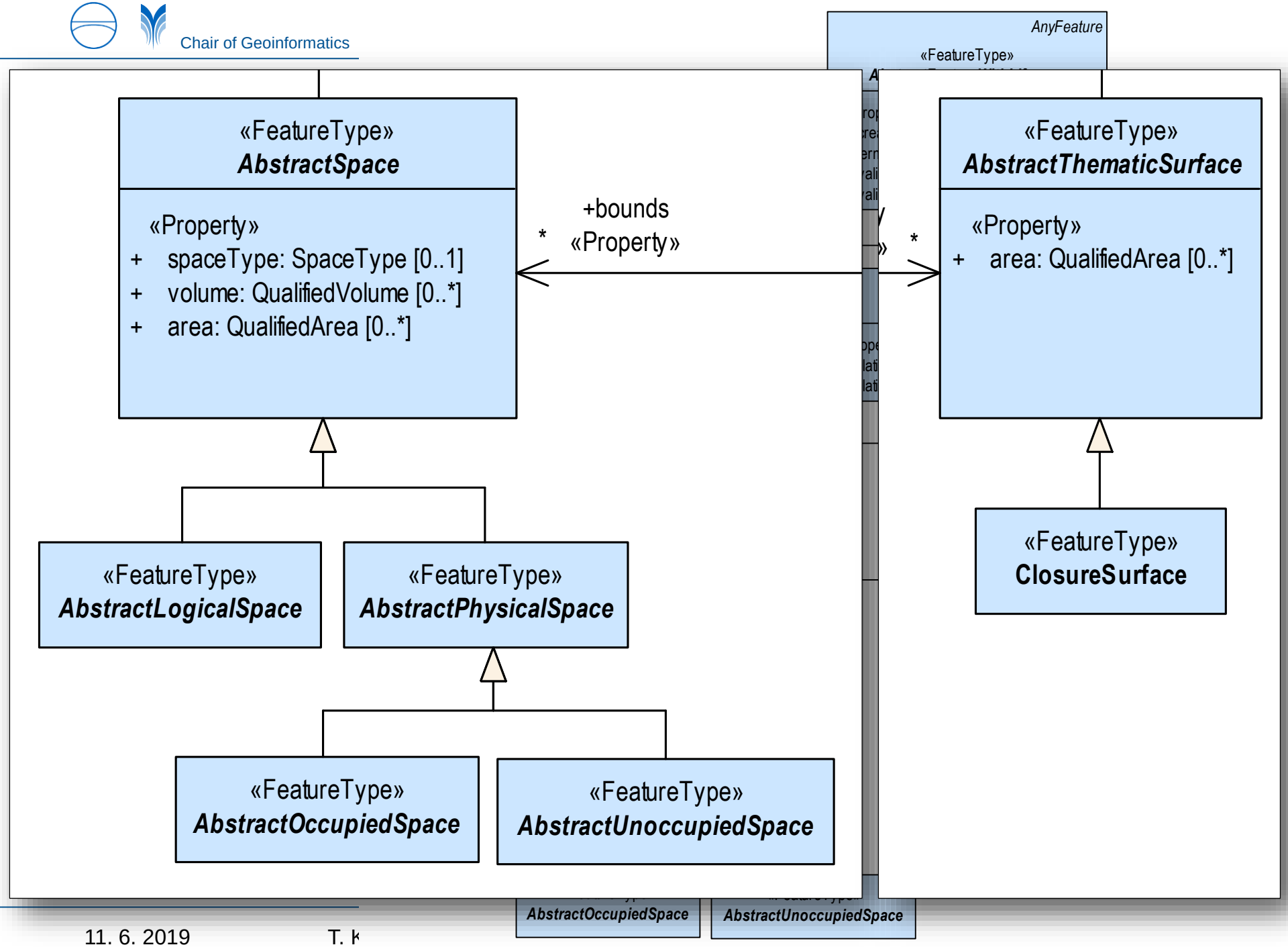
LOD4 – Architectural models (interior)
- "Walkable" architectural models

- CityGML 3.0 allows for representing the interior of buildings, tunnels and bridges in LODs 0-3 as well.
  → E.g., the exterior can now be modelled in LOD1, whereas the interior is represented in LOD2 or 3
- Supports the use of 3D city models in applications which require detailed representations of the indoor, but not necessarily of the outdoor, e.g. indoor navigation and energy applications.

Revised Core Module – Introduction of a space concept

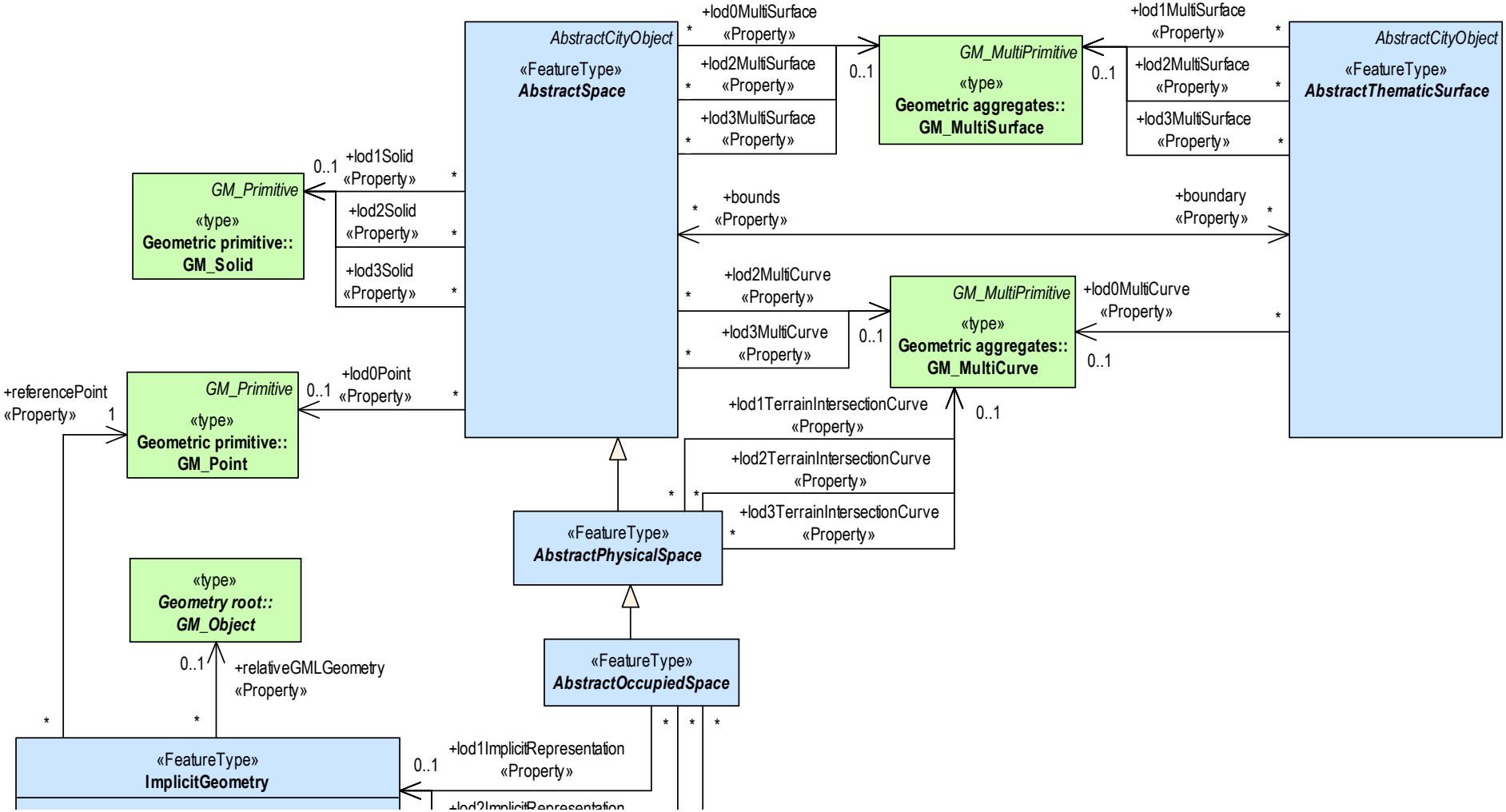
- All thematic objects are now either spaces or space boundaries by basing them on the two pivotal abstract classes *AbstractSpace* and *AbstractThematicSurface*.



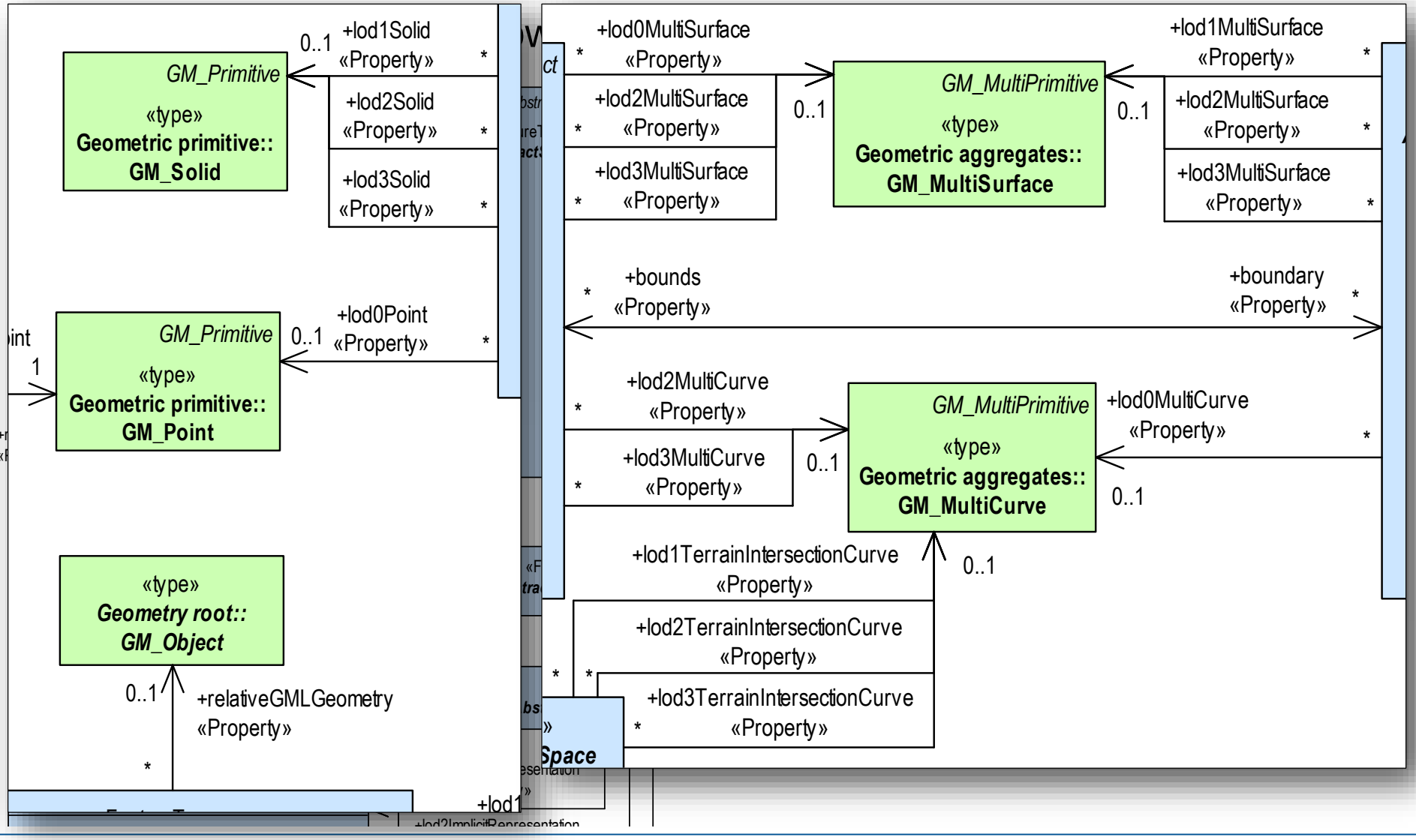

«FeatureType»
AbstractSpace
«Property»
+ spaceType: SpaceType [0..1]
+ volume: QualifiedVolume [0..*]
+ area: QualifiedArea [0..*]
+bounds
«Property»
*
*
«FeatureType»
AbstractThematicSurface
«Property»
+ area: QualifiedArea [0..*]
«FeatureType»
AbstractLogicalSpace
«FeatureType»
AbstractPhysicalSpace
«FeatureType»
ClosureSurface
«FeatureType»
AbstractOccupiedSpace
«FeatureType»
AbstractUnoccupiedSpace

Revised Core Module – Geometry and LOD Concept

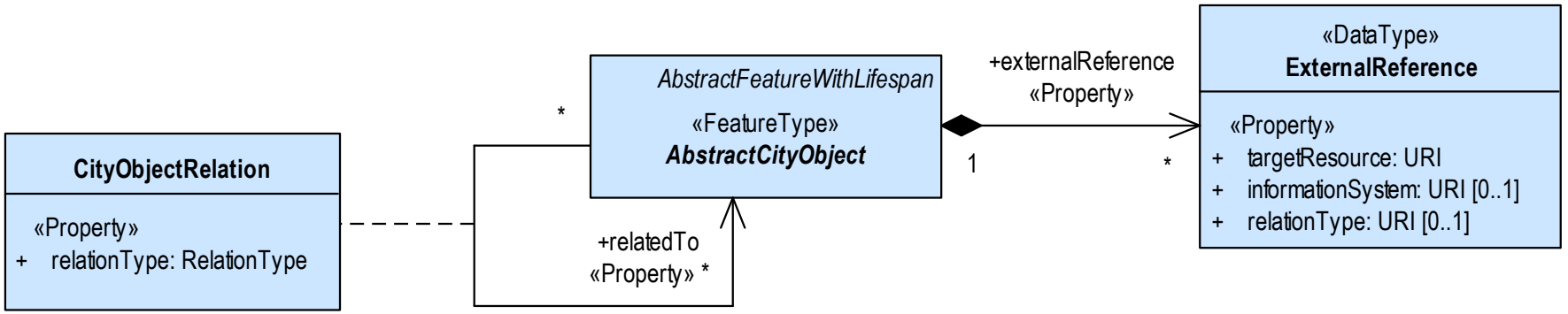
- The LOD concept is now part of the Core module → less redundancy

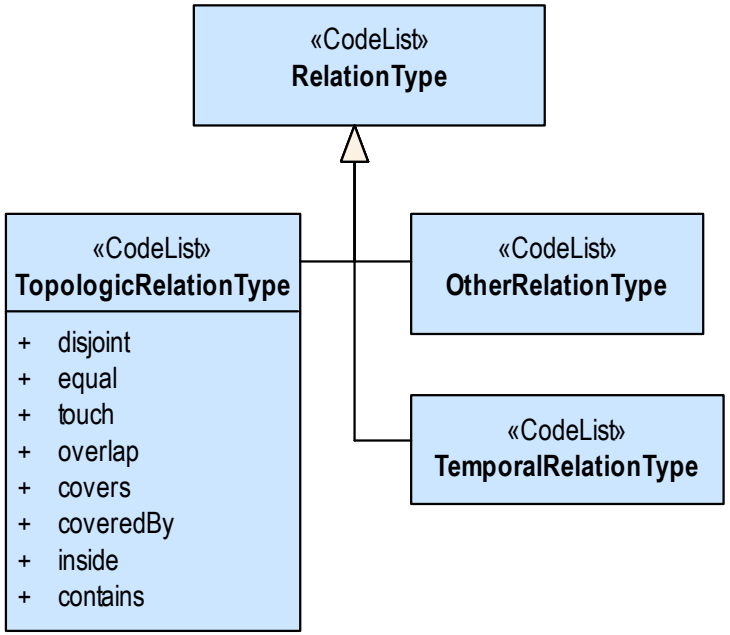


Revised Core Module – Geometry and LOD Concept



Revised Core Module – City object relations

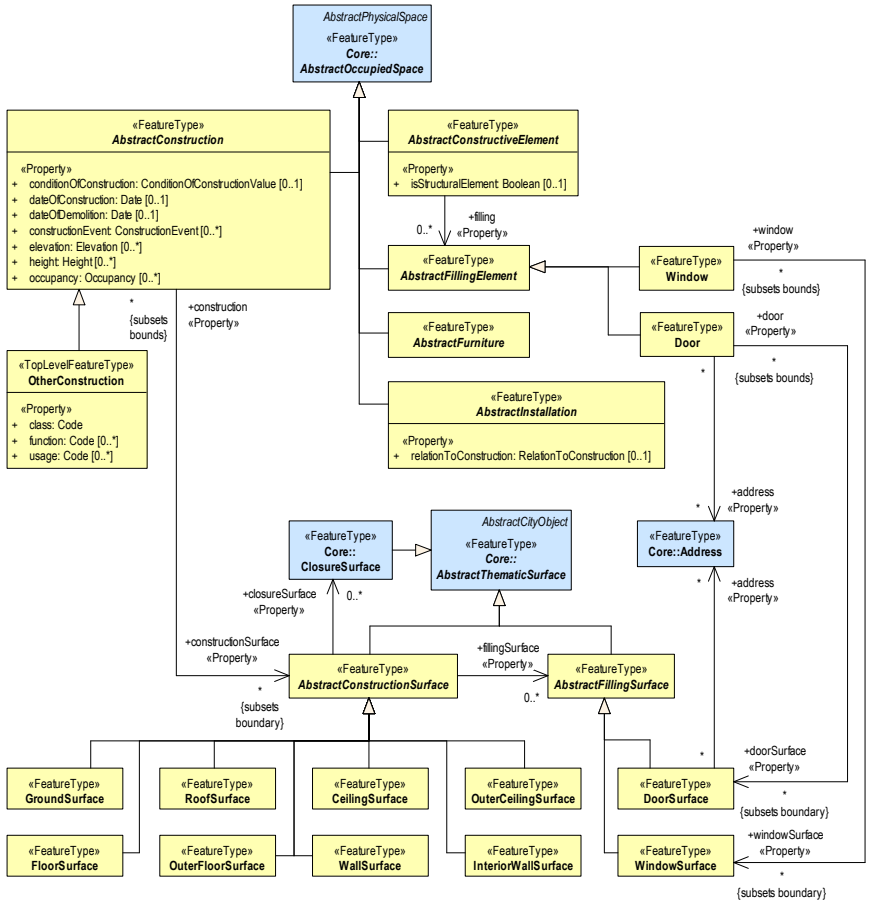




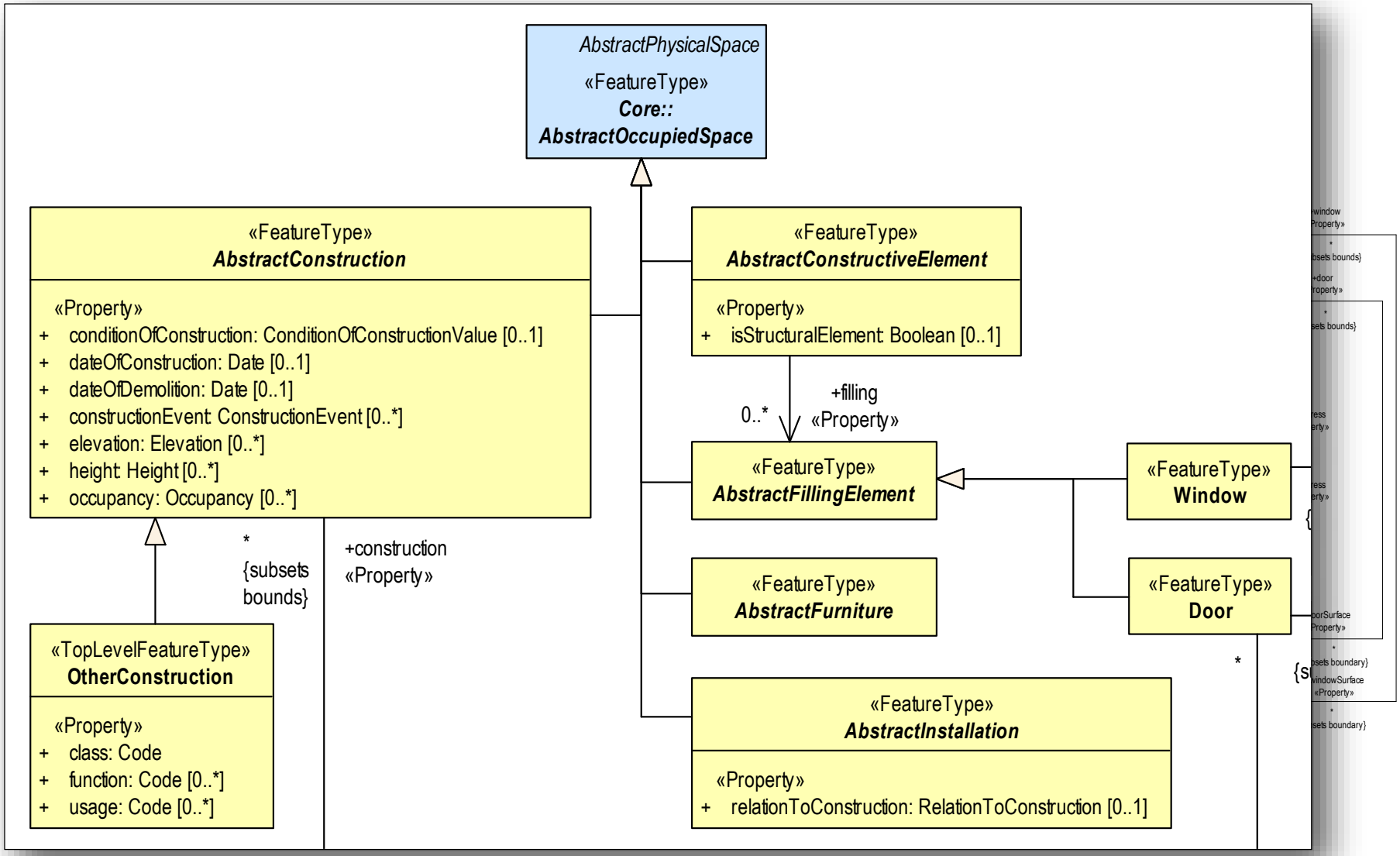
- The new class *CityObjectRelation* allows for defining arbitrary relations between any city objects.
- The class *ExternalReference* is now better aligned to an RDF representation. References can be additionally qualified by a relation type (e.g. the sameAs relation from OWL) → allows for mapping to RDF triples.

New Construction module

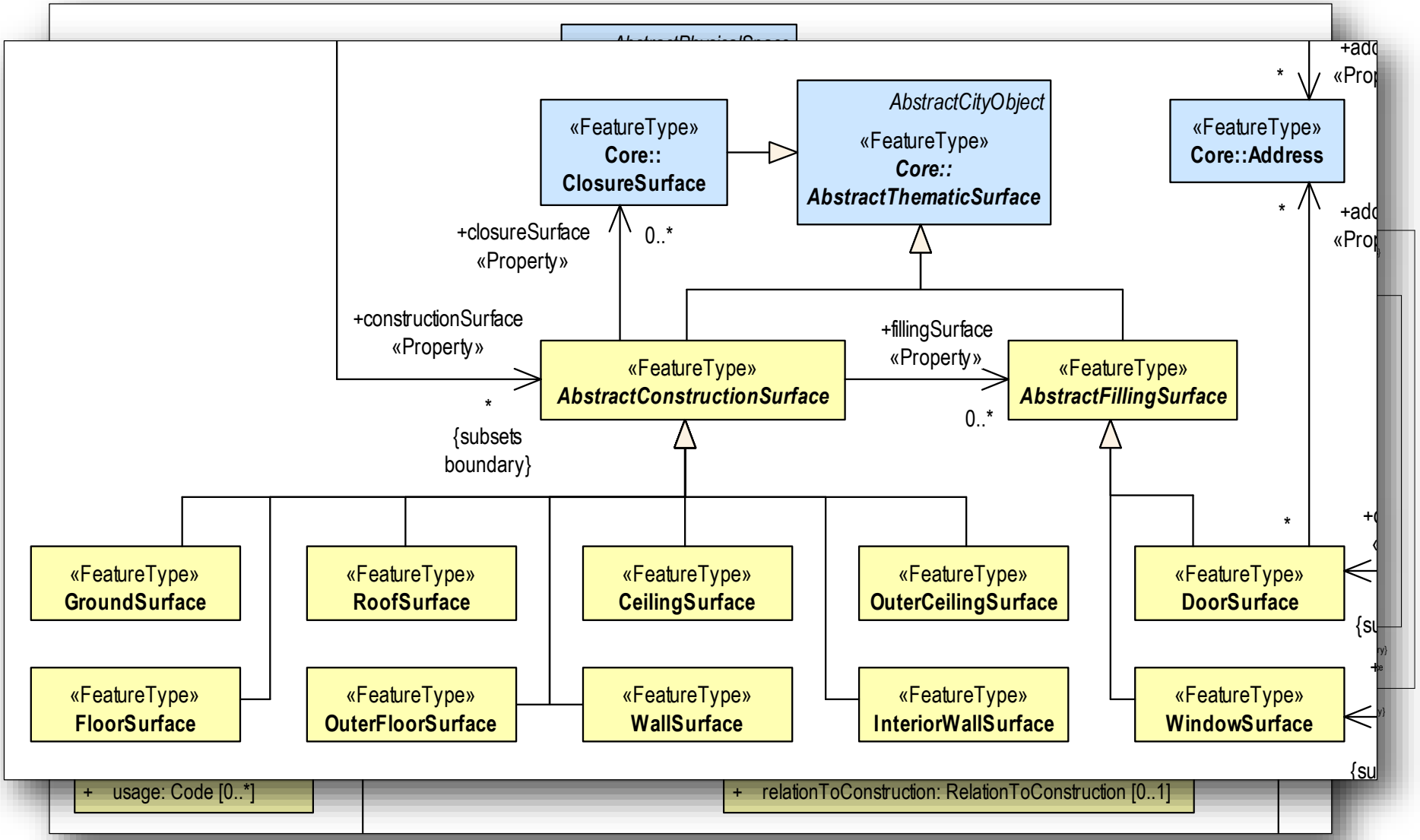
- Defines concepts common to all kinds of man-made constructions
  - Integrates all classes which are similar over different types of constructions like buildings, tunnels, and bridges.
- Introduces a new class *OtherConstruction* to represent constructions which are neither buildings, tunnels, nor bridges.
- Introduces a new class *AbstractConstructiveElement* for better interoperability with IFC.
- Improved definition of elevation levels and measured heights.



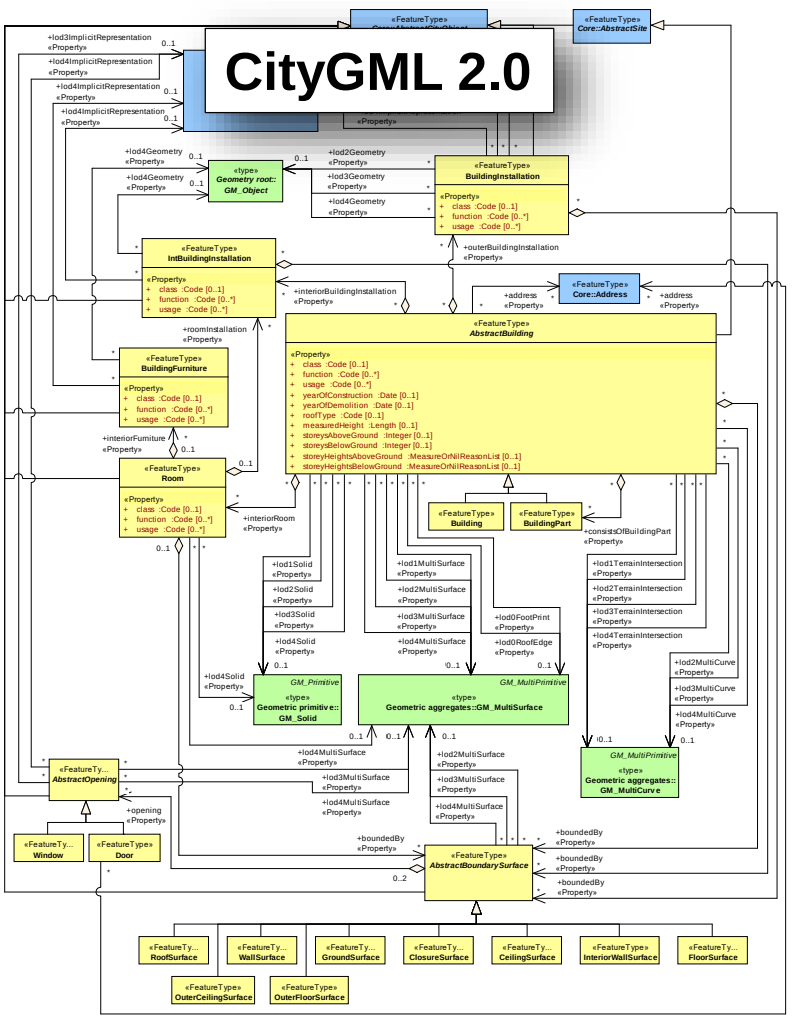
New Construction module

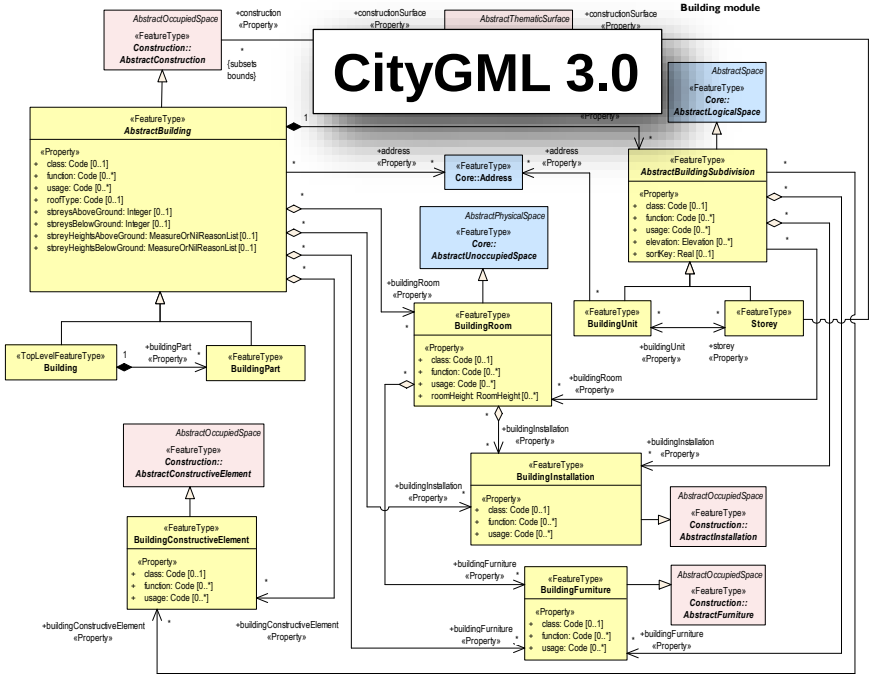


New Construction module



Revised Building module



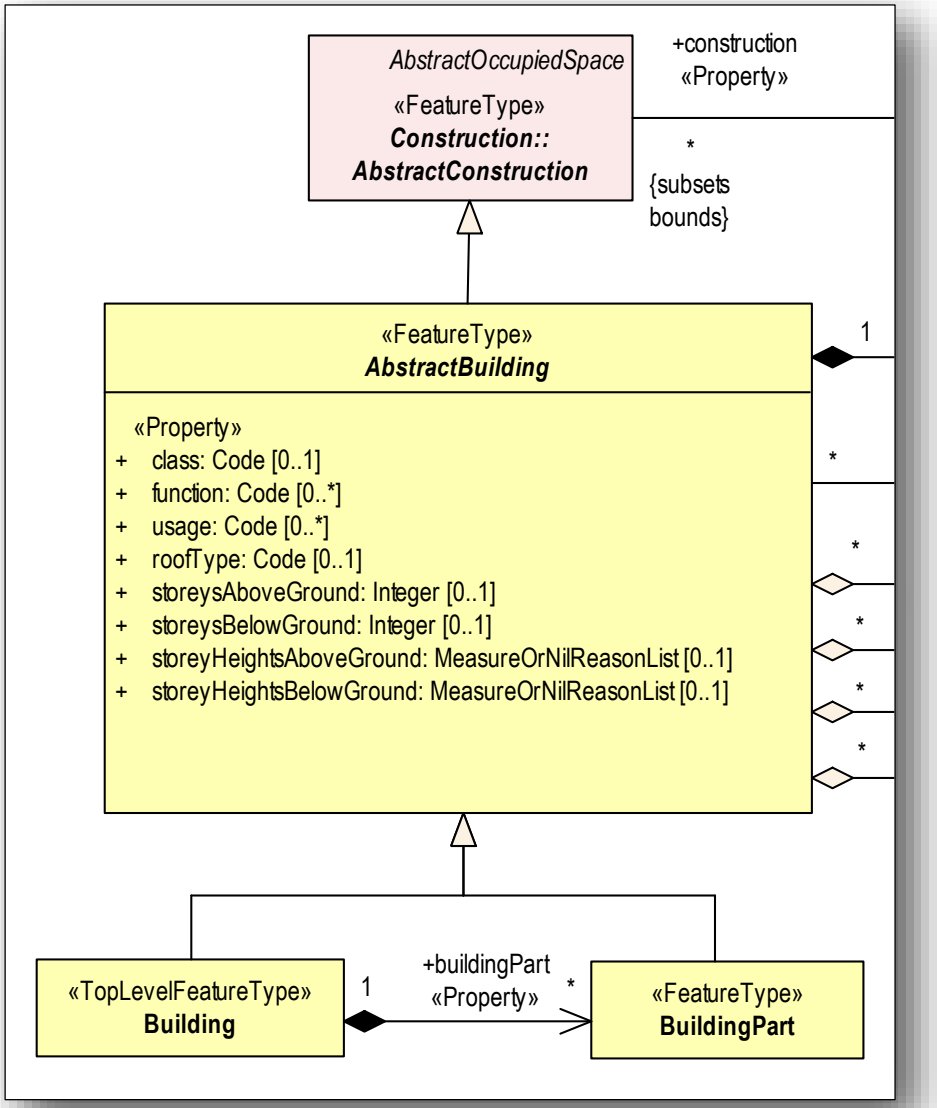


- Geometries are now defined in the Core module
- Thematic surfaces and openings are now defined in the Construction module

→ Inheritance of these concepts reduces the size of the Building module

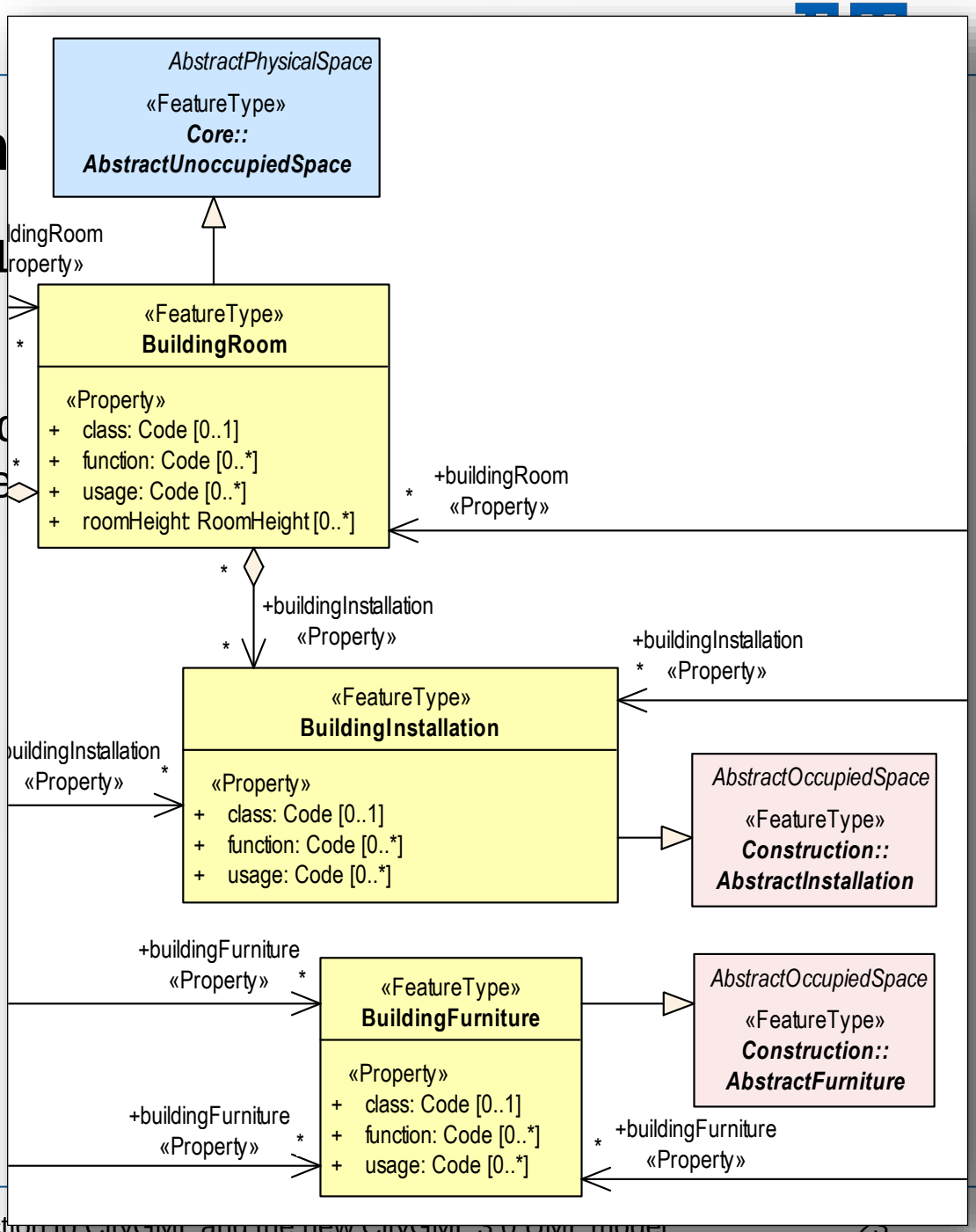
Revised Building module

- All concepts from CityGML 2.0 are preserved.
  They are now subclasses of the basic concepts defined in the Construction and Core modules.



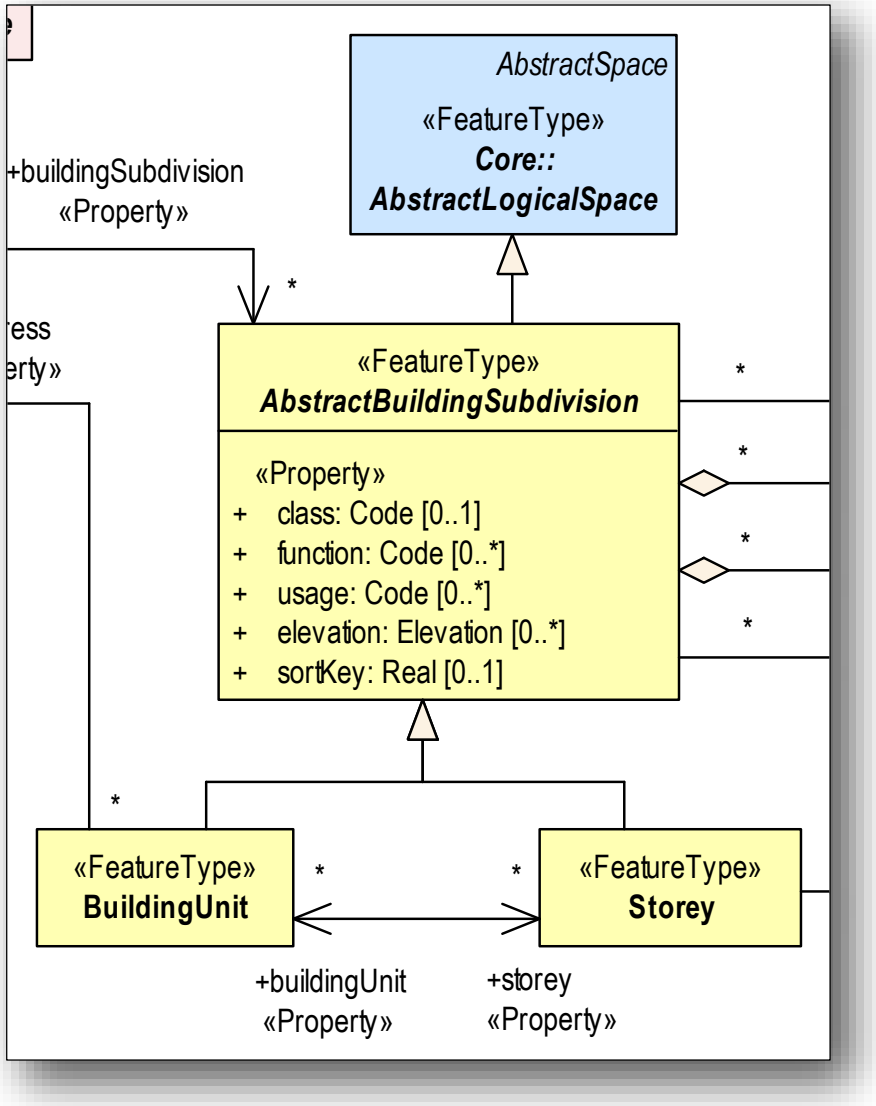
Revised Building m

- All concepts from CityGML are preserved.
  They are now subclasses the basic concepts defined the Construction and Core modules.



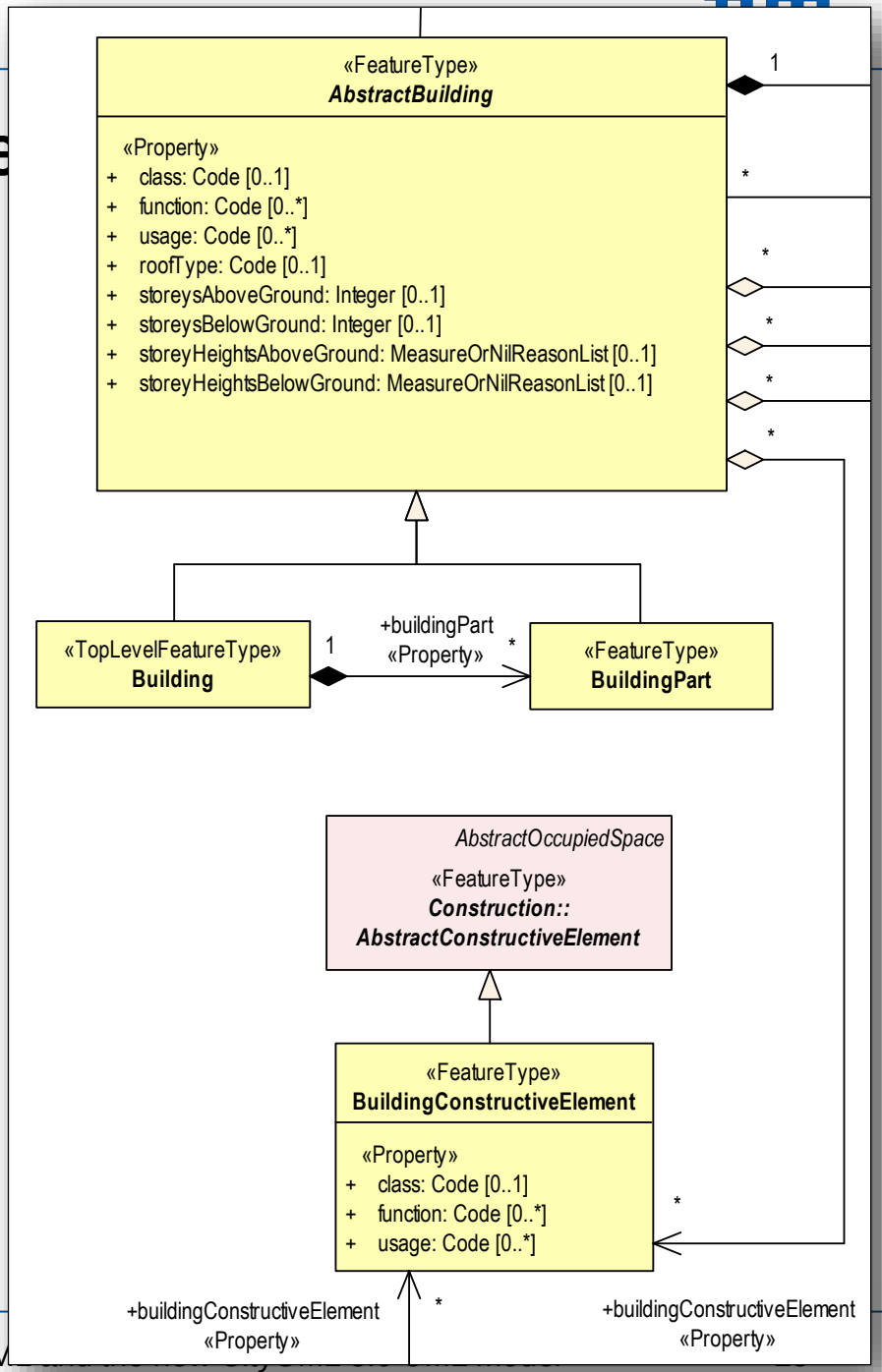
Revised Building module

- All concepts from CityGML 2.0 are preserved.
  They are now subclasses of the basic concepts defined in the Construction and Core modules.
- Introduction of a new class *AbstractBuildingSubdivision* to allow for representing building units and storeys.



Revised Building module

- All concepts from CityGML 2.0 are preserved.
  They are now subclasses of the basic concepts defined in the Construction and Core modules.
- Introduction of a new class *AbstractBuildingSubdivision* to allow for representing building units and storeys.
- Introduction of the class *BuildingConstructiveElement* facilitates interoperability with IFC.

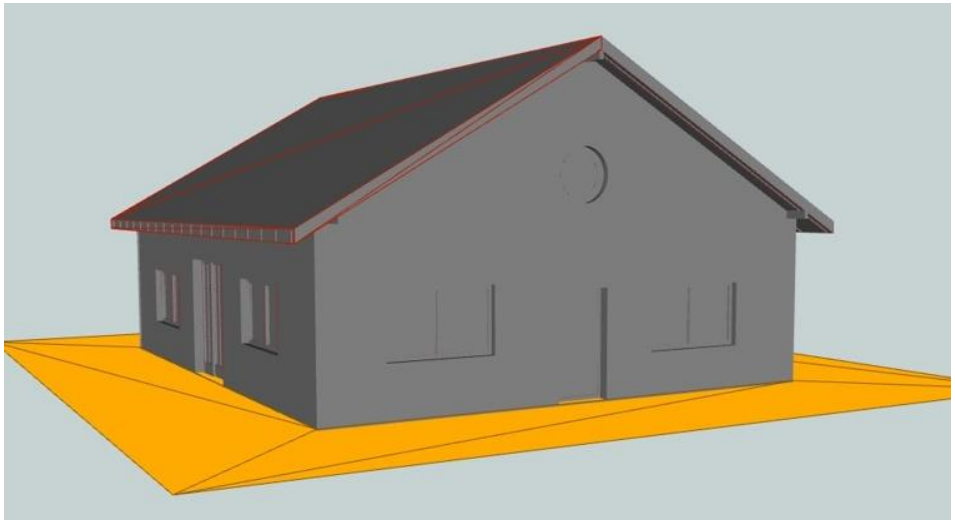


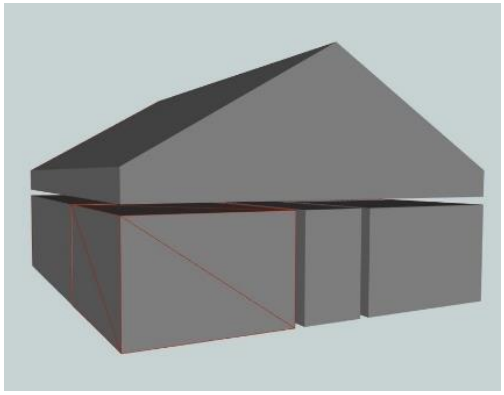
Improved IFC – CityGML Interoperability

Example: Semantic Mapping of an IFC building to CityGML 3.0

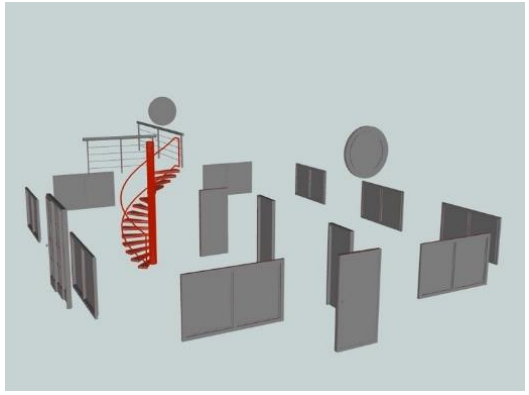
IFC objects		CityGML objects
IfcProject	→	CityModel
IfcSite	→	LandUse
IfcBuilding	→	Building
IfcBuildingStorey	→	**Storey**
IfcSpace	→	BuildingRoom
IfcWallStandardCase	→	**BuildingConstructiveElement**
IfcBeam, IfcSlab, IfcMember	→	**BuildingConstructiveElement**
IfcDoor, IfcWindow	→	**BuildingConstructiveElement**
IfcRailing, IfcStair	→	BuildingInstallation

The FZK House represented in CityGML 3.0

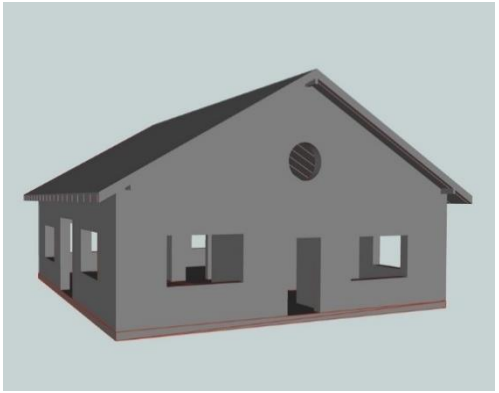




Rooms



Building Installations,
Doors and Windows



BuildingConstructiveElements

Changes in the context of semantic 3D city models

- **Highly dynamic changes**
  - Spatial properties: e.g. moving objects
  - Thematic properties: e.g. energy consumption
  - Appearance: e.g. raster images showing air quality
  - Real-time data from Sensors and IoT devices

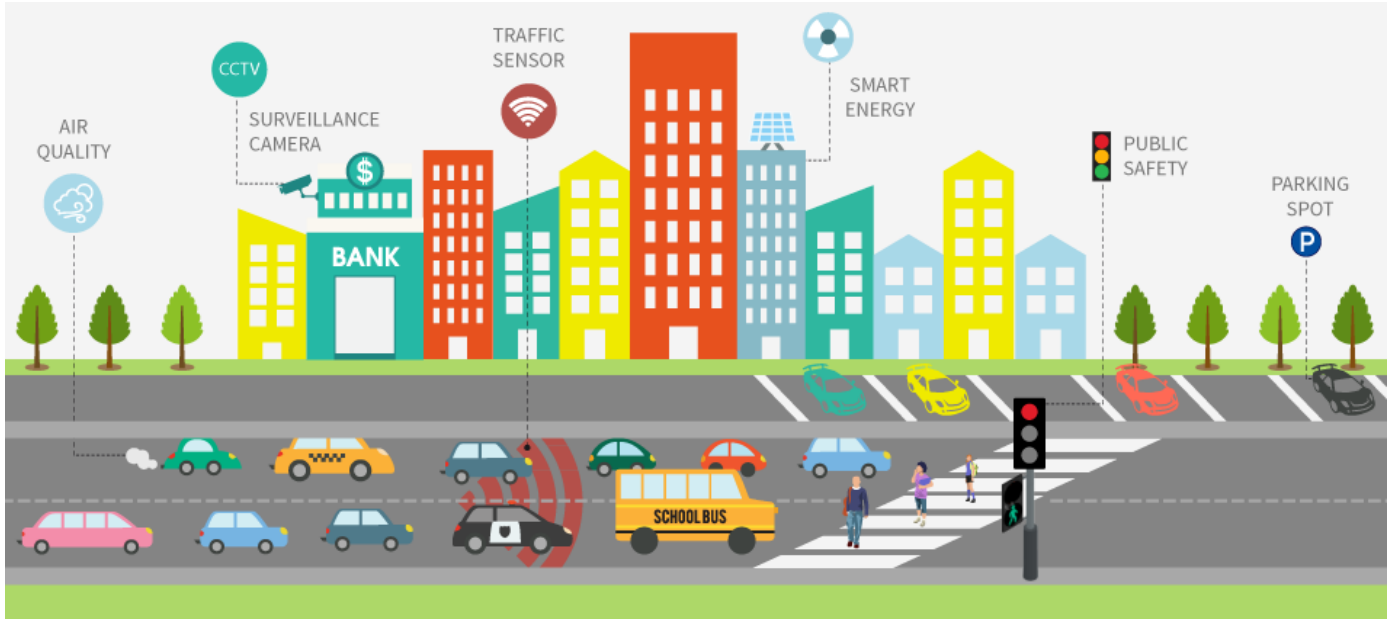
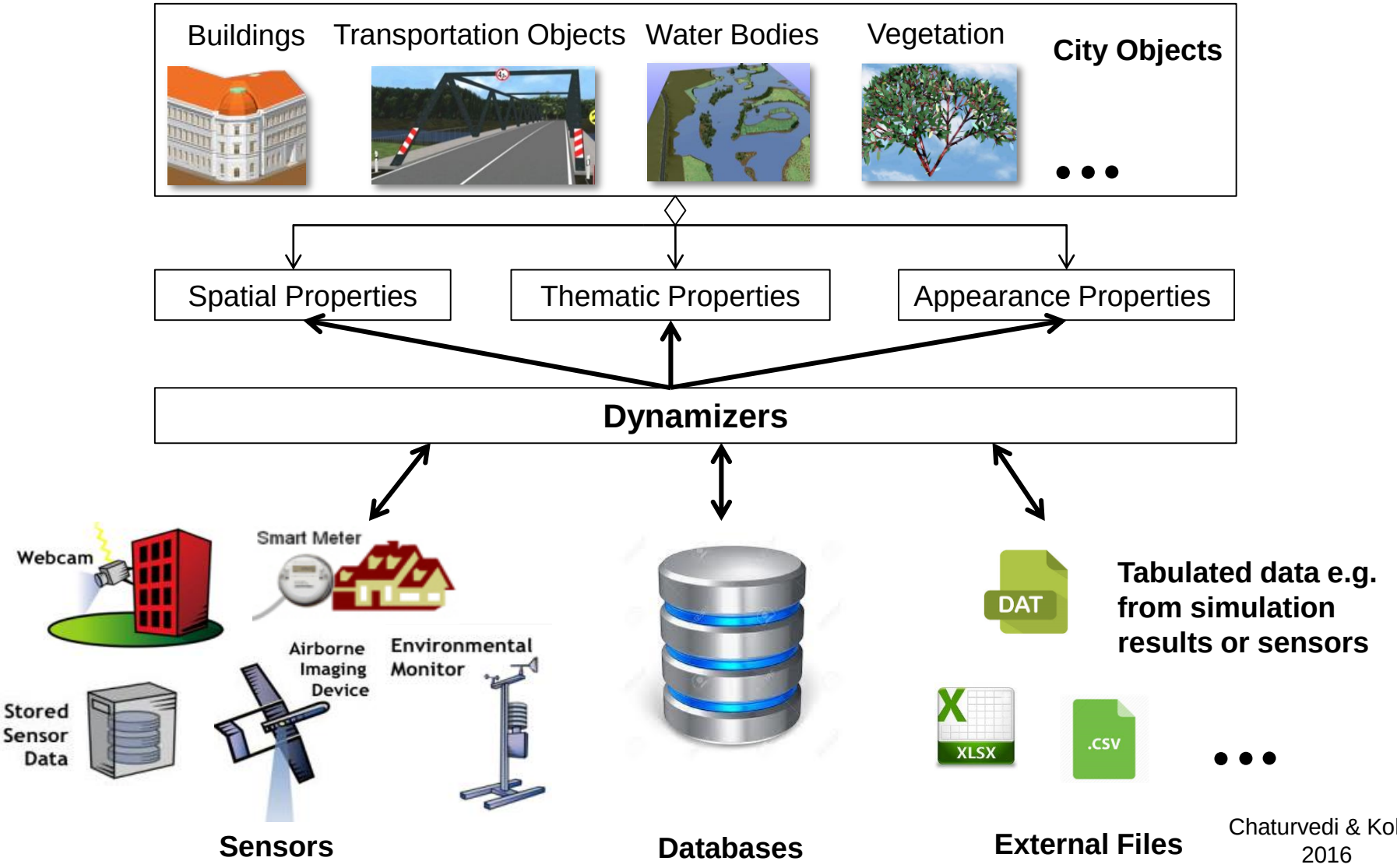


Image Source: www.houseofbots.com

New Dynamizer module



Chaturvedi & Kolbe 2016

New Dynamizer module – Example Scenario

CityGML object

```
<bldg:WallSurface gml:id = "UUID_01_WS_1">
     <gen:doubleAttribute name = "globalRadMonth">
          <gen:value = xxx />
     </gen:doubleAttribute>
  </Building>
```

One dynamic attribute which changes with time

Replacing dynamic attributes using XPath

Simulation Results

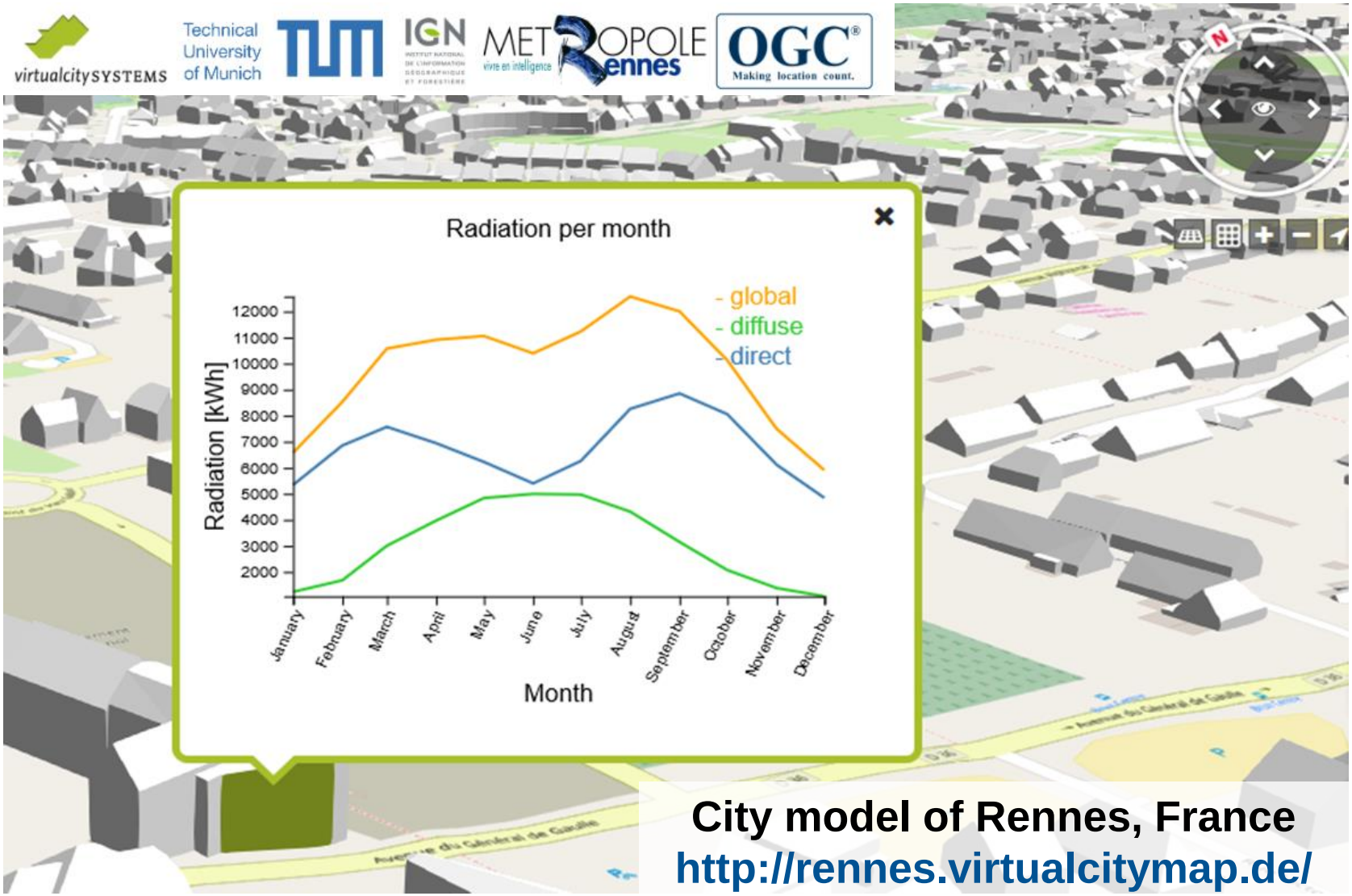
Month	Global Radiation
JAN-15	4293.446
FEB-15	5563.502
MAR-15	7010.33
. . .	. . .
DEC-15	4010.239

```
<cityObjectMember>
  <dyn:Dynamizer>
    <dyn:attributeRef> //bldg:WallSurface [@gml:id = 'UUID_01_WS_1']
                       /doubleAttribute[@name = 'globalRadMonth']
                       /gen:value
    </dyn:attributeRef>
    <dyn:startTime> 2015-01-01T00:00:00Z </dyn:startTime>
    <dyn:endTime> 2015-12-31T00:00:00Z </dyn:endTime>
    <dyn:dynamicData>.. </dyn:dynamicData>
  </dyn:Dynamizer>
</cityObjectMember>
```

Dynamizer

Respresenting data in standardized ways, such as OGC TimeseriesML, OGC Observations & Measurements

Dynamizer in OGC Future City Pilot Phase 1



City model of Rennes, France
http://rennes.virtualcitymap.de/

Linking to Sensors in standardized ways

Sensor (PV Panel)

building1_roofSurface1

building1

```
<cityObjectMember>
  <dyn:Dynamizer gml:id = "PV_Power_Timeseries" >
    <dyn:attributeRef>//RoofSurface[@gml:id ='building1_roofSurface1']
                      /doubleAttribute[@name =  'PV_Power']
                      /gen:value </dyn:attributeRef>
    <dyn:startTime>2016-01-01T00:00:00Z</startTime>
    <dyn:endTime>2016-12-01T00:00:00Z</endTime>
    <dyn:linkToSensor>
     <dyn:SensorConnection>
      <dyn:sensorID>. . . </dyn:sensorID>
      <dyn:serviceType>. . . </dyn:serviceType>
      <dyn:linkToObservation>. . . </dyn:linkToObservation>
      <dyn:linkToSensorML>. . . </dyn:linkToSensorML>
      <dyn:sensorLocation xlink:href="#building1_roofSurface1"/>
     </dyn:SensorConnection>
    </dyn:linkToSensor>
  </dyn:Dynamizer>
<cityObjectMember>
```

Unique Sensor ID

SOS or SensorThings API

SOS GetObservation

SOS DescribeSensor

Link to CityGML Object

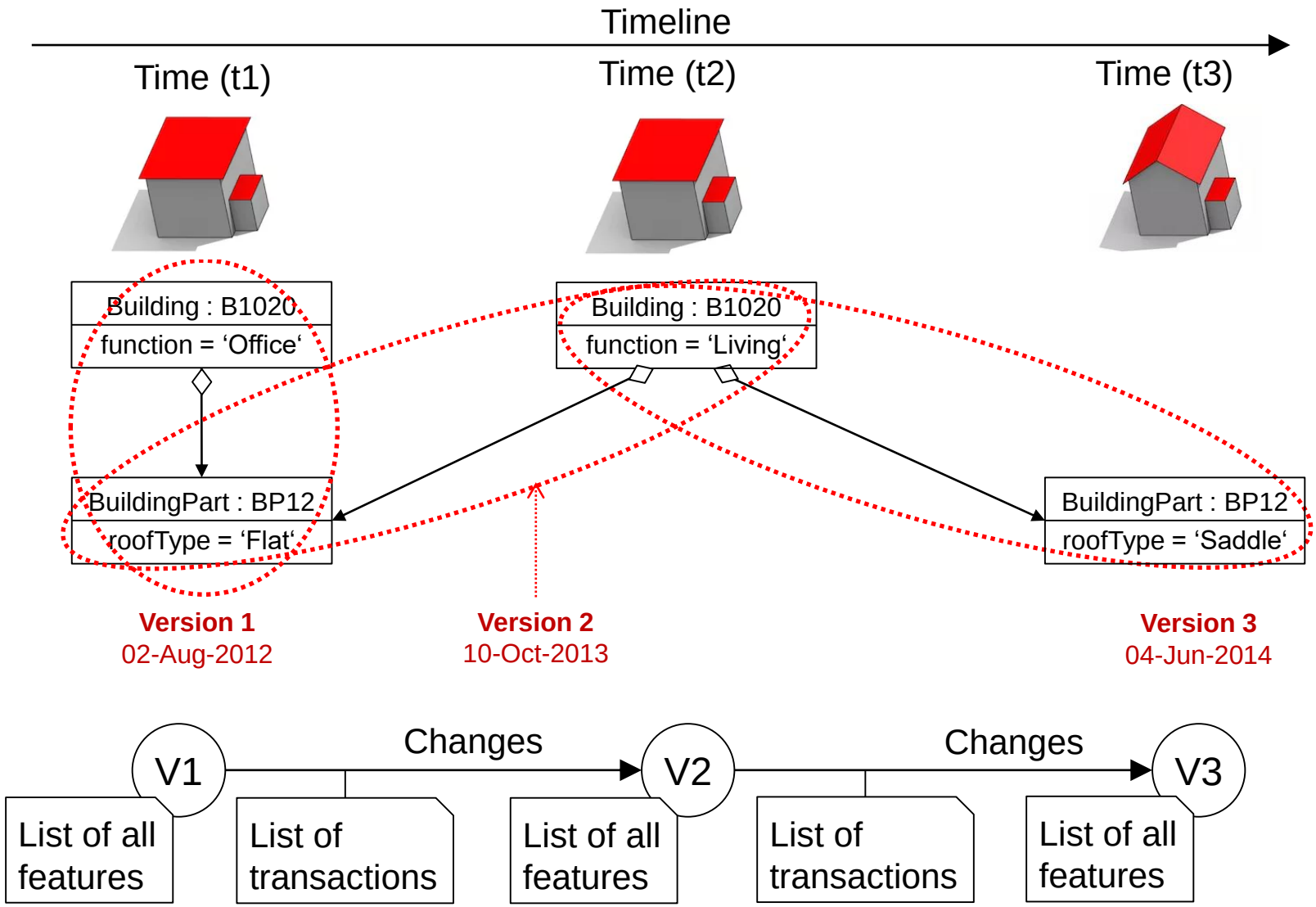
Image source : http://www.royalgreengas.com/index.php/photovoltaic/residential-buildings

Changes in the context of semantic 3D city models

- **Slower changes**
  - History or evolution of cities/city models
  - Change of feature‘s geometry over time
  - Managing parallel or alternative versions over time
- New **Versioning Module:**
  - Explicit modeling of changes
  - Snapshots of city models at a specific point in time
  - All objects can have bitemporal lifespan data (date of creation / termination, valid from / valid to).
  - Multiple versions of city objects can be represented within one city model data set.

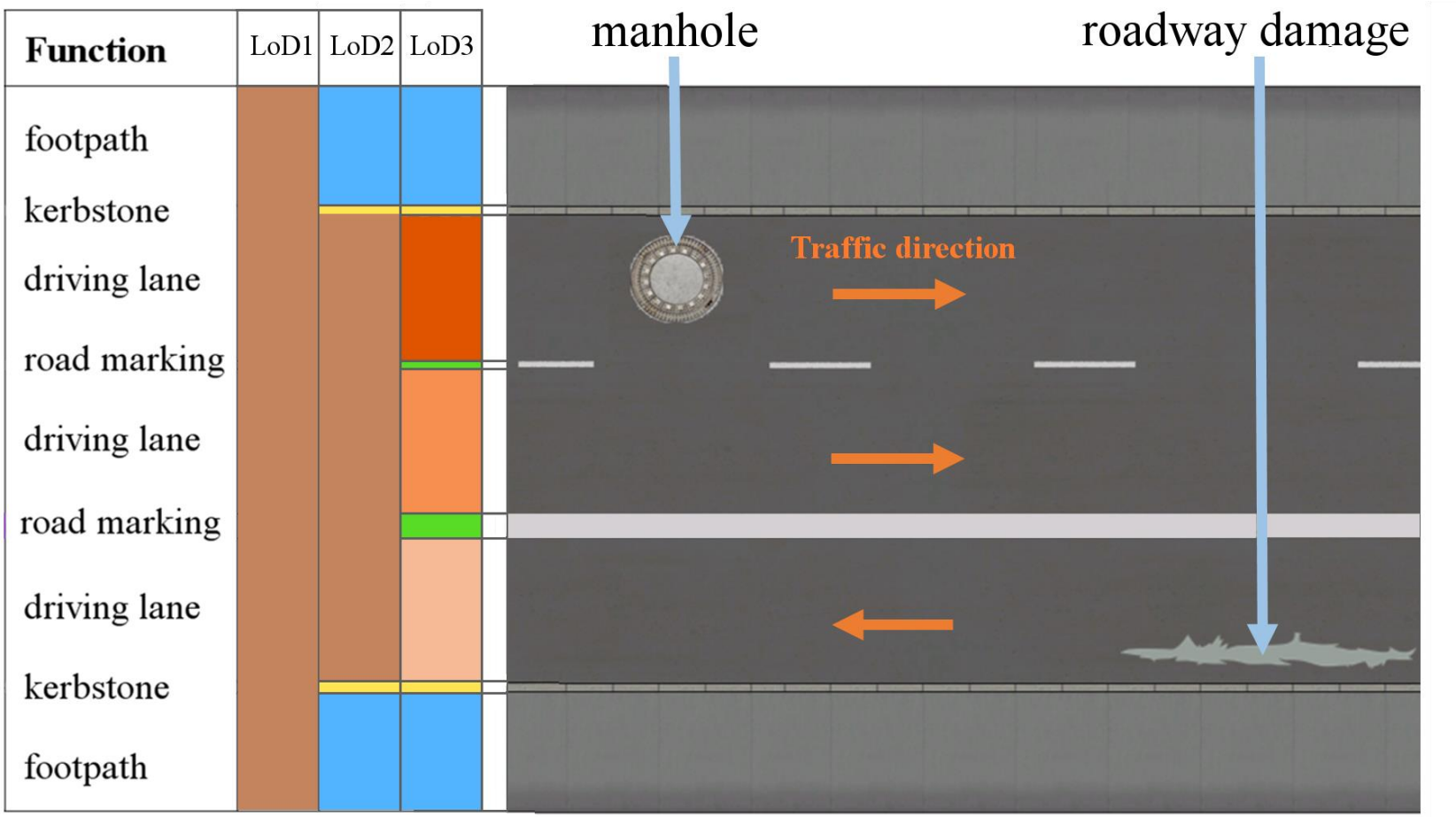
Image Source: www.pinterest.com

New Versioning module – Example Scenario



Revised Transportation module

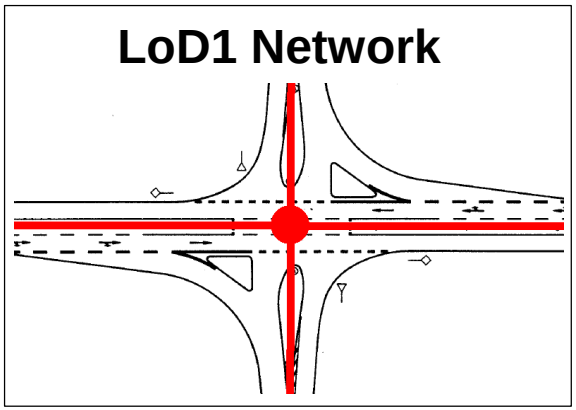
- Improved LOD-based modelling of street space objects

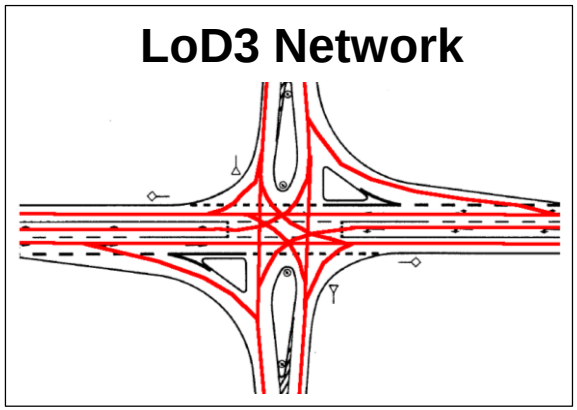


Beil & Kolbe, 2017

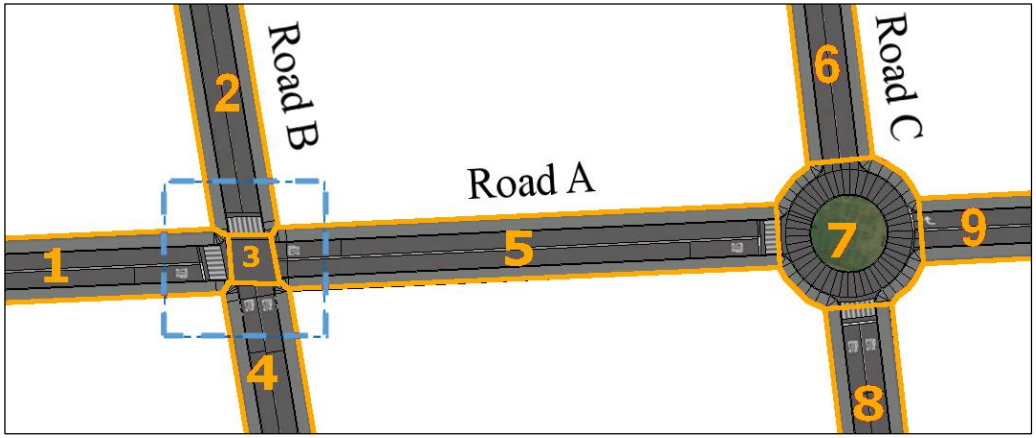
Revised Transportation module

- Introduction of linear / graph representation in LOD 1-3



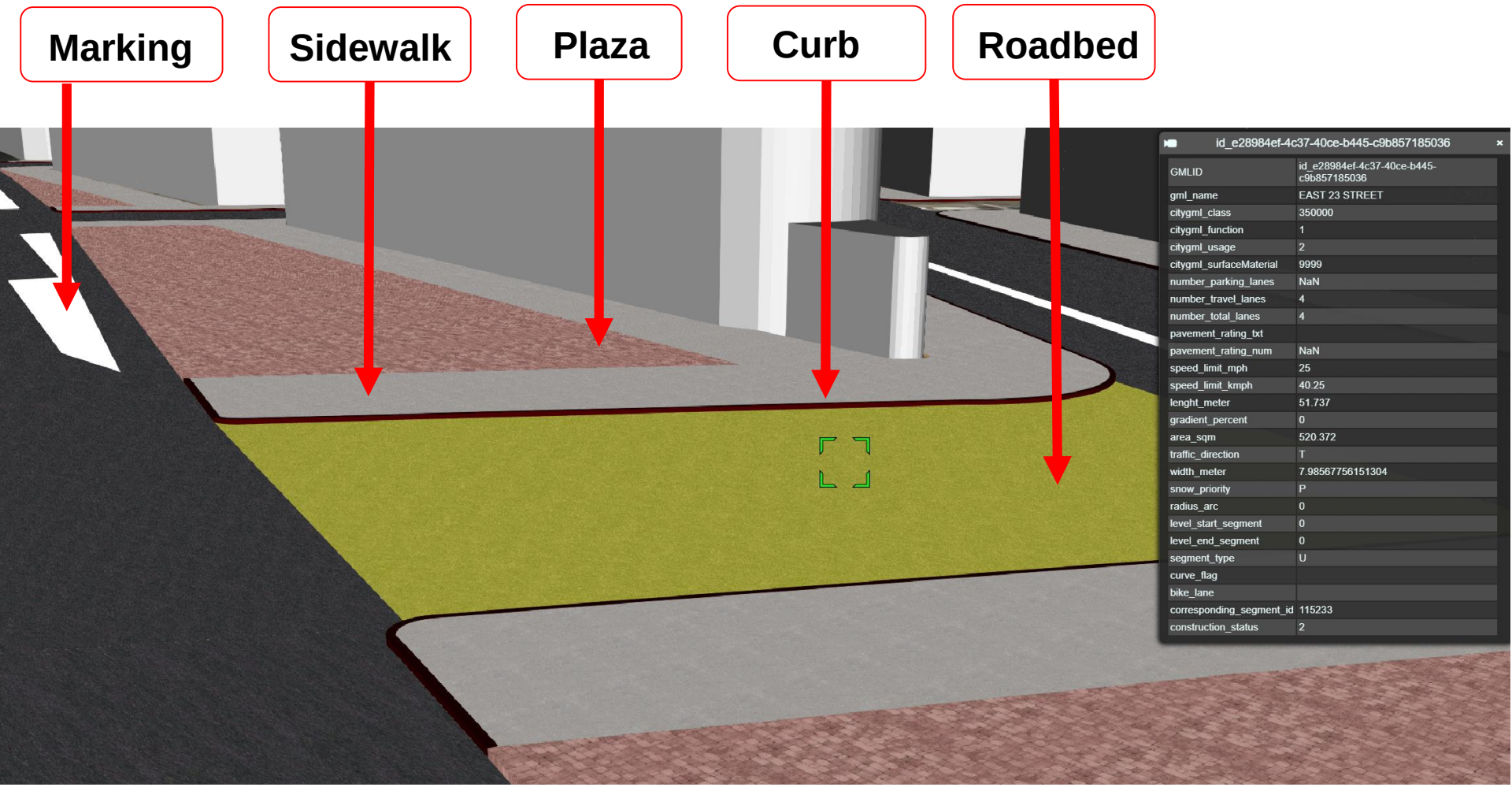


- Introduction of sections
- One Section per
  - Road Segment
  - Intersection
  - Roundabout
  - Dead End

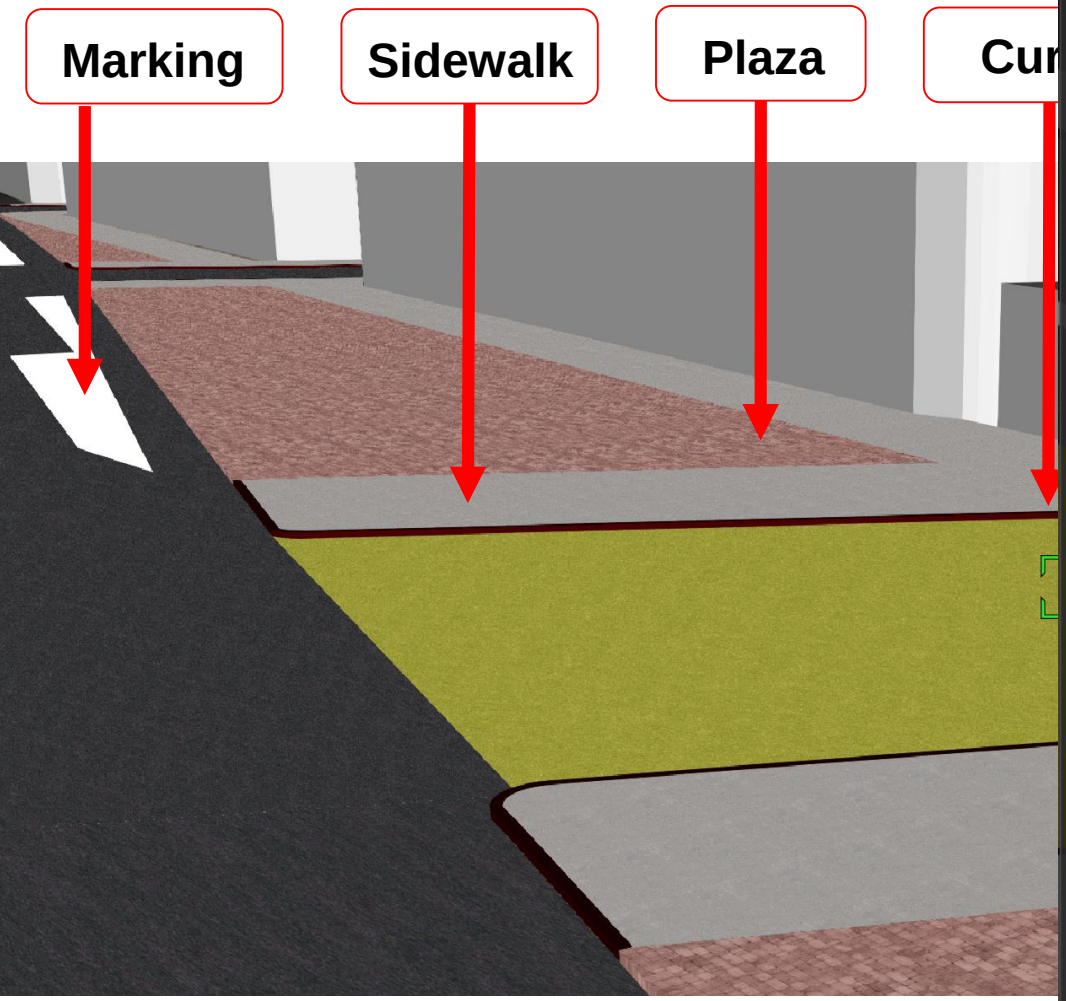


Beil & Kolbe, 2017

Demonstration of Results – 3D Visualization



Demonstration of Results



id_e28984ef-4c37-40ce-b445-c9b857185036

GMLID	id_e28984ef-4c37-40ce-b445-c9b857185036
gml_name	EAST 23 STREET
citygml_class	350000
citygml_function	1
citygml_usage	2
citygml_surfaceMaterial	9999
number_parking_lanes	NaN
number_travel_lanes	4
number_total_lanes	4
pavement_rating_txt	
pavement_rating_num	NaN
speed_limit_mph	25
speed_limit_kmph	40.25
lenght_meter	51.737
gradient_percent	0
area_sqm	520.372
traffic_direction	T
width_meter	7.98567756151304
snow_priority	P
radius_arc	0
level_start_segment	0
level_end_segment	0
segment_type	U
curve_flag	
bike_lane	
corresponding_segment_id	115233
construction_status	2

Conclusions (I)

- New LOD concept allows for more flexibility in representing the interior and exterior of city objects + representing objects as point clouds
- Revised Core module and new Construction module provide
  - better interoperability with other standards (IndoorGML, IFC, RDF, LADM, INSPIRE)
  - simplification of geometry handling
  - improved representation of physical and logical objects
- New and revised modules improve the use of 3D city models in different areas of application (urban planning, energy simulations, traffic analyses, …)
- State-of-the art UML model generation including ISO-compliance + automatic derivation of exchange formats
  - The CityGML XML Schemas are derived fully automatically from the UML model

Conclusions (II)

- All software tools that are able to read and process generic GML 3 application schemas can work directly with CityGML 3 application schemas as well
  (e.g. FME, HALE, GDAL, Interactive Instruments WFS, Degree, GALDOS WFS, CPA SupportGIS)
- However: Everything is still subject to final voting within the OGC!

Resources

- CityGML 3.0 UML Diagrams
  https://github.com/opengeospatial/CityGML-3.0CM

- CityGML 3.0 XML Schema Files
  https://github.com/opengeospatial/CityGML-3.0Encodings

- Open Source Conversion Tool CityGML 2.0 → CityGML 3.0 (currently Building module only)
  https://github.com/tum-gis/citygml2-to-citygml3

- IFC → CityGML 3.0 FME Workspace
  https://github.com/tum-gis/ifc-to-citygml3