Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2021-01-28>

External identifier of this OGC® document: http://www.opengis.net/doc/IS/CityGML-2/3.0

Internal reference number of this OGC® document: YY-nnnrx

Version: 0.1

Category: OGC® Implementation Specification

Editor: Charles Heazel

OGC City Geography Markup Language (CityGML) Part 2: GML Encoding Standard

Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Standard
Document subtype:	Encoding Specification
Document stage:	Draft
Document language:	English

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

Table of Contents

i. Abstract

The CityGML 3.0 GML Encoding Standard presents the implementation-dependent, GML encoding of the concepts defined by the CityGML 3.0 Conceptual Model(CM) standard. Those concepts include the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical, and appearance properties. “City” is broadly defined to comprise not just built structures, but also elevation, vegetation, water bodies, city furniture, and more. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties.

CityGML is an implementation of the CityGML 3.0 Conceptual Model Standard. Table 1 maps requirements classes from the CityGML conceptual model into the implementation details documented in this standard.

Table 1. Conceptual Model Mapping

Conceptual Model	**Section**	**GML Schema**
ADE	Base Conformance Class	cityGMLBase.xsd
Appearance	Appearance Conformance Class	appearance.xsd
Bridge	Bridge Conformance Class	bridge.xsd
Building	Building Conformance Class	building.xsd
City Furniture	City Furniture Conformance Class	cityFurniture.xsd
City Object Group	City Object Group Conformance Class	cityObjectGroup.xsd
Construction	Construction Conformance Class	construction.xsd
Core	Base Conformance Class	cityGMLBase.xsd
Dynamizer	Dynamizer Conformance Class	dynamizer.xsd
Generics	Generics Conformance Class	generics.xsd
Land Use	Land Use Conformance Class	landUse.xsd
Point Cloud	Point Cloud Conformance Class	pointCloud.xsd
Relief	Relief Conformance Class	relief.xsd
Transportation	Transportation Conformance Class	transportation.xsd
Tunnel	Tunnel Conformance Class	tunnel.xsd
Vegetation	Vegetation Conformance Class	vegetation.xsd
Versioning	Versioning Conformance Class	versioning.xsd
Water Body	Water Body Conformance Class	waterBody.xsd

ii. Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, CityGML, 3D city models, GML, XML

iii. Preface

In order to achieve consensus on the basic entities, attributes, and relations of a 3D city model, a UML Conceptual Model, CityGML 3.0, was approved as an OGC standard in March, 2021. This model provides a unifying conceptual basis for city model encoding standards. This cityGML 3.0 XML Encoding Standard defines how those concepts should be realized using XML and GML technologies.

As an OGC standard, CityGML follows the OGC modular specification standard, OGC 08-131r3. Because of the breadth of CityGML, its conceptual model was divided into separate Requirements Classes, one for each subject area. This CityGML encoding similarly is divided into Requirements Classes which are then grouped into Parts. A Part may address multiple CityGML Requirements Classes but each Requirements Class is addressed in a single part. Because Requirements Classes may depend on other Requirements Classes the reader of this CityGML Part may need to conform to Requirements Classes in other Parts as well.

Note that this CityGML encoding standard is a standardization target of the CityGML 3.0 Conceptual Model Standard. Therefore this standard conforms to the Conformance Classes in that standard. Evidence of that conformance is provided in Conceptual Model Conformance (Normative). On the other hand, an application claiming conformance to this CityGML encoding standard must conform to the Requirements Classes contained in this standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Table 2. Submitting Organizations

Organization	Points of Contact
Heazeltech LLC	Charles Heazel

v. Submitters

All questions regarding this submission should be directed to the editor or the submitters:

Table 3. Submission Contact Points

Name	Institution
Charles Heazel	Heazeltech LLC

Chapter 1. Scope

This Standard documents the OGC GML Implementation Specification (IS) for the CityGML 3.0 Conceptual Model. The CityGML 3.0 conceptual model is a Platform Independent Model (PIM). It defines concepts in a manner which is independent of any implementing technology. As such, the CityGML CM cannot be implemented directly. Rather, it serves as the base for Platform Specific Models (PSM). A PSM adds to the PIM the technology-specific details needed to fully define the CityGML model for use with a specific technology. The PSM can then be used to generate the schema and other artifacts needed to build CityGML 3.0 implementations.

This standard defines the PSMs and schemas for the CityGML 3.0 Implementation Specification (IS) for Geography Markup Language (GML) implemenations.

Chapter 2. Conformance

This standard defines an Implementation Specification which specifies how the CityGML 3.0 Conceptual Model should be implemented using Geography Markup Language (GML). The Standardization Target for this standard is:

1. Implementations of the CityGML 3.0 Conceptual Model using GML encodings.

2.1. Implementation Specifications

Implementation Specifications define how a Conceptual Model should be implemented using a specific technology. Conformant Implementation Specifications provide evidence that they are an accurate representation of the Conceptual Model. This evidence includes data demonstrating that the applicable criteria documented in the CityGML 3.0 CM Abstract Test Suite have been satisfied. That evidence is provided in Conceptual Model Conformance (Normative).

2.2. Conformance Classes

This standard identifies seventeen (17) conformance classes. One conformance class is defined for each GML schema. Each conformance class is defined by one requirements class. The tests in Annex A are organized by Requirements Class. So an implementation of the *Base* conformance class must pass all tests specified in Annex A for the *Base* requirements class.

Of these seventeen conformance classes, only the *Base* conformance class is mandatory. All other conformance classes are optional. In the case where a conformance class has a dependency on another conformance class, that conformance class should also be implemented.

Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of OGC TBD. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of OGC TBD are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

- IETF: RFC 2045 & 2046, Multipurpose Internet Mail Extensions (MIME). (November 1996),
- IETF: RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. (January 2005)
- INSPIRE: D2.8.III.2 Data Specification on Buildings – Technical Guidelines. European Commission Joint Research Centre.
- ISO: ISO 19101-1:2014, Geographic information - Reference model - Part 1: Fundamentals
- ISO: ISO 19103:2015, Geographic Information – Conceptual Schema Language
- ISO: ISO 19105:2000, Geographic information – Conformance and testing
- ISO: ISO 19107:2003, Geographic Information – Spatial Schema
- ISO: ISO 19108:2002/Cor 1:2006, Geographic information – Temporal schema — Technical Corrigendum 1
- ISO: ISO 19109:2015, Geographic Information – Rules for Application Schemas
- ISO: ISO 19111:2019, Geographic information – Referencing by coordinates
- ISO: ISO 19123:2005, Geographic information — Schema for coverage geometry and functions
- ISO: ISO 19156:2011, Geographic information – Observations and measurements
- ISO: ISO/IEC 19505-2:2012, Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure
- ISO/IEC 19507:2012, Information technology — Object Management Group Object Constraint Language (OCL)
- ISO: ISO/IEC 19775-1:2013 Information technology — Computer graphics, image processing and environmental data representation — Extensible 3D (X3D) — Part 1: Architecture and base components
- Khronos Group Inc.: COLLADA – Digital Asset Schema Release 1.5.0
- OASIS: Customer Information Quality Specifications - extensible Address Language (xAL), Version v3.0
- OGC: The OpenGIS® Abstract Specification Topic 5: Features, OGC document 08-126
- OGC: The OpenGIS™ Abstract Specification Topic 8: Relationships Between Features, OGC document 99-108r2
- OGC: The OpenGIS™ Abstract Specification Topic 10: Feature Collections, OGC document 99-110

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Standard.

For the purposes of this document, the following additional terms and definitions apply.

2D data
geometry of features is represented in a two-dimensional space
NOTE In other words, the geometry of 2D data is given using (X,Y) coordinates.
[INSPIRE D2.8.III.2, definition 1]

2.5D data
geometry of features is represented in a three-dimensional space with the constraint that, for each (X,Y) position, there is only one Z
[INSPIRE D2.8.III.2, definition 2]

3D data
Geometry of features is represented in a three-dimensional space.
NOTE In other words, the geometry of 2D data is given using (X,Y,Z) coordinates without any constraints.
[INSPIRE D2.8.III.2, definition 3]

application schema
A set of conceptual schema for data required by one or more applications. An application schema contains selected parts of the base schemas presented in the ORM Information Viewpoint. Designers of application schemas may extend or restrict the types defined in the base schemas to define appropriate types for an application domain. Application schemas are information models for a specific information community.
OGC Definitions Register at http://www.opengis.net/def/glossary/term/ApplicationSchema

codelist
A value domain including a code for each permissible value.

conceptual model
model that defines concepts of a universe of discourse
[ISO 19101-1:2014, 4.1.5]

conceptual schema

1. formal description of a conceptual model
   [ISO 19101-1:2014, 4.1.6]
2. base schema. Formal description of the model of any geospatial information. Application schemas are built from conceptual schemas.
   OGC Definitions Register at http://www.opengis.net/def/glossary/term/ConceptualSchema

Implementation Specification

Specified on the OGC Document Types Register at http://www.opengis.net/def/doc-type/is

levels of detail
quantity of information that portrays the real world
NOTE The concept comprises data capturing rules of spatial object types, the accuracy and the types of geometries, and other aspects of a data specification. In particular, it is related to the notions of scale and resolution.
[INSPIRE Glossary]

life-cycle information
set of properties of a spatial object that describe the temporal characteristics of a version of a spatial object or the changes between versions
[INSPIRE Glossary]

Platform (Model Driven Architecture)
the set of resources on which a system is realized.
[Object Management Group, Model Driven Architecture Guide rev. 2.0]

Platform Independent Model
a model that is independent of a spacific platform
[Object Management Group, Model Driven Architecture Guide rev. 2.0]

Platform Specific Model
a model of a system that is defined in terms of a specific platform
[Object Management Group, Model Driven Architecture Guide rev. 2.0]

Chapter 5. Conventions

5.1. Identifiers

The normative provisions in this document are denoted by the URI

http://www.opengis.net/spec/CityGML-2/3.0

All requirements and conformance tests that appear in this document are denoted by partial URIs relative to this base.

5.2. UML Notation

This standard is an implementation of the CityGML Conceptual Model (CM) Standard. The CityGML conceptual model was constructed using the Unified Modeling Language (UML). Exerpts from that model appear in this standard. The UML notations used are described in the diagram in UML notation (see ISO TS 19103, Geographic information - Conceptual schema language)..

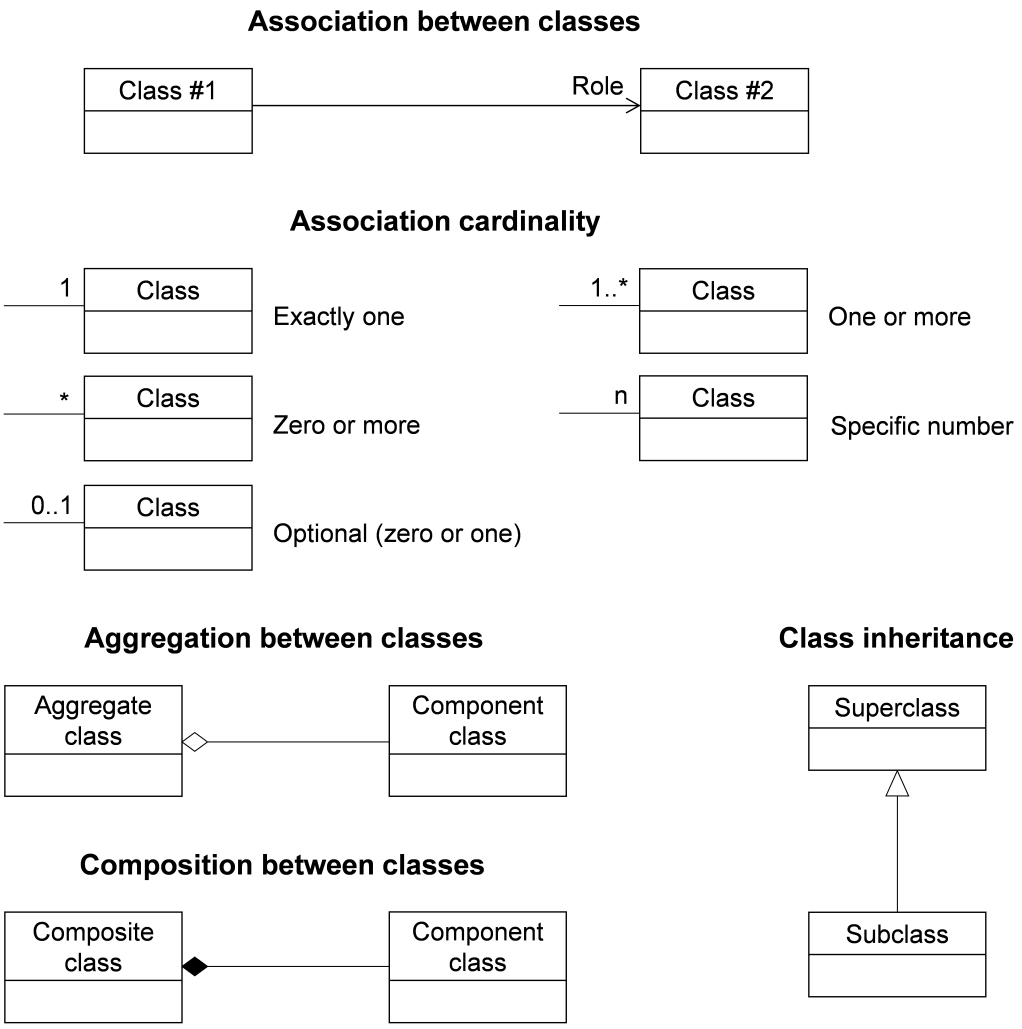


Figure 1. UML notation (see ISO TS 19103, Geographic information - Conceptual schema language).

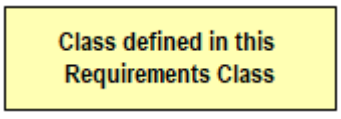
All associations between model elements in the CityGML Conceptual Model are uni-directional. Thus, associations in the model are navigable in only one direction. The direction of navigation is

depicted by an arrowhead. In general, the context an element takes within the association is indicated by its role. The role is displayed near the target of the association. If the graphical representation is ambiguous though, the position of the role has to be drawn to the element the association points to.

The following stereotypes are used in the model:

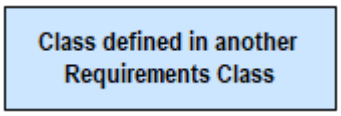
- «ApplicationSchema» denotes a conceptual schema for data required by one or more applications. In the CityGML Conceptual Model, every module is defined as a separate application schema to allow for modularization.
- «FeatureType» represents features that are similar and exhibit common characteristics. Features are abstractions of real-world phenomena and have an identity.
- «TopLevelFeatureType» denotes features that represent the main components of the conceptual model. Top-level features may be further semantically and spatially decomposed and substructured into parts.
- «Type» denotes classes that are not directly instantiable, but are used as an abstract collection of operation, attribute and relation signatures. The stereotype is used in the CityGML Conceptual Model only for classes that are imported from the ISO standards 19107, 19109, 19111, and 19123.
- «ObjectType» represents objects that have an identity, but are not features.
- «DataType» defines a set of properties that lack identity. A data type is a classifier with no operations, whose primary purpose is to hold information.
- «Enumeration» enumerates the valid attribute values in a fixed list of named literal values. Enumerations are specified in the CityGML Conceptual Model.
- «BasicType» defines a basic data type.
- «CodeList» enumerates the valid attribute values. In contrast to Enumeration, the list of values is open and, thus, not given inline in the CityGML UML Model. The allowed values can be provided within an external code list.
- «Union» is a list of attributes. The semantics are that only one of the attributes can be present at any time.
- «Property» denotes attributes and association roles. This stereotype does not add further semantics to the conceptual model, but is required to be able to add tagged values to the attributes and association roles that are relevant for the encoding.
- «Version» denotes that the value of an association role that ends at a feature type is a specific version of the feature, not the feature in general.

In order to enhance the readability of the CityGML UML diagrams, classes are depicted in different colors. The following coloring scheme is applied:

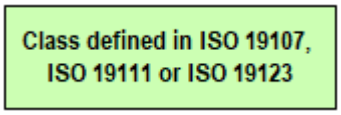


Classes painted in yellow belong to the Requirements Class which is subject of discussion in that clause of the standard in which the UML diagram is given. For example, in the context of

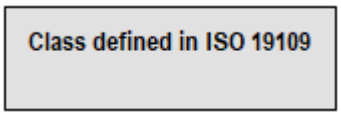
[rc_core_section], which introduces the *CityGML Core* module, the yellow color is used to denote classes that are defined in the *CityGML Core* Requirements Class. Likewise, the yellow classes shown in the UML diagram in [rc_building-model_section] are associated with the *Building* Requirements Class that is subject of discussion in that chapter.



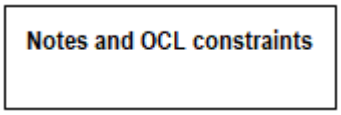
Classes painted in blue belong to a Requirements Class different to that associated with the yellow color. In order to explicitly denote to which Requirements Class these classes belong, their class names are preceded by the UML package name of that Requirements Class. For example, in the context of the *Building* Requirements Class, classes from the *CityGML Core* and the *Construction* Requirements Classes are painted in blue and their class names are preceded by *Core* and *Construction*, respectively.



Classes painted in green are defined in the ISO standards 19107, 19111, or 19123. Their class names are preceded by the UML package name, in which the classes are defined.



Classes painted in grey are defined in the ISO standard 19109. In the context of this standard, this only applies to the class *AnyFeature*. *AnyFeature* is an instance of the metaclass *FeatureType* and acts as super class of all classes in the CityGML UML model with the stereotype «FeatureType». A metaclass is a class whose instances are classes.



The color white is used for notes and Object Constraint Language (OCL) constraints that are provided in the UML diagrams.

The example UML diagram in Example UML diagram demonstrating the UML notation and coloring scheme used throughout the CityGML Standard. demonstrates the UML notation and coloring scheme used throughout this standard. In this example, the yellow classes are associated with the *CityGML Building* module, the blue classes are from the *CityGML Core* and *Construction* modules, and the green class depicts a geometry element defined by ISO 19107.

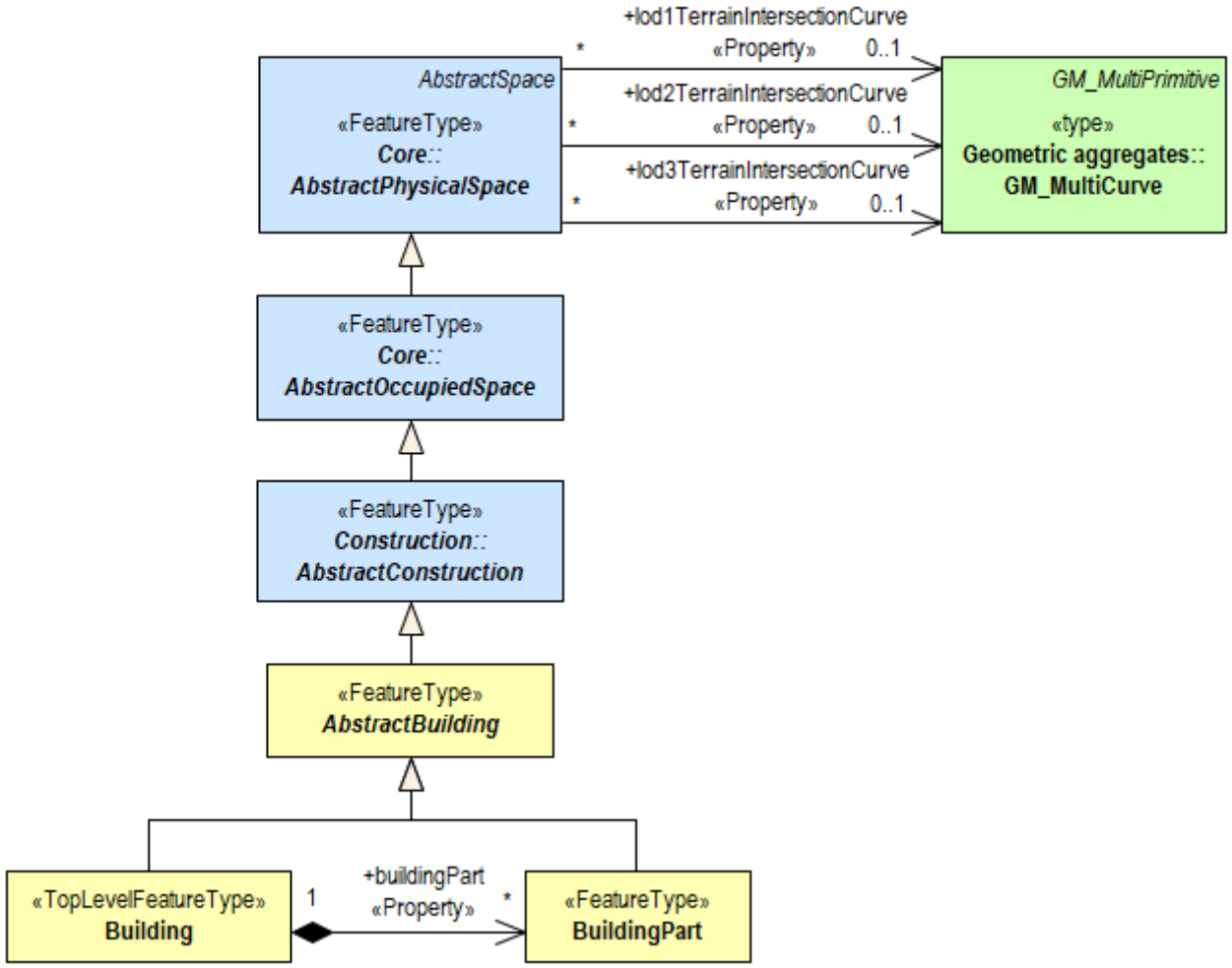


Figure 2. Example UML diagram demonstrating the UML notation and coloring scheme used throughout the CityGML Standard.

5.3. XML Notation

TBD

Chapter 6. Requirements

6.1. Base Conformance Class

The Base Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Core: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-core
- ADE: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-ade

The applicable GML schema is cityGMLBase.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base	
Target type	Implementation
Dependency	cityGMLBase.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-core
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-ade

Requirement 1	**/req/base/elements**
A conforming application shall support the CityGML XML elements listed in [base-xml-elements] in accordance with the GML XML schema specified in cityGMLBase.xsd.	

6.2. Appearance Conformance Class

NOTE	replace with appropriate text

Features of the land, such as naturally occurring water features and vegetation are specified in the LandFeature Requirements Class as land features. Also included are models of the land surface and subsurface layers. Improvements to the land such as the construction of an embankment or the planting of landscape material are considered to be part of Site Facilities in the Facility Requirements Class.

The Appearance Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Appearance: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-appearance

The applicable GML schema is appearance.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-appearance	
Target type	Implementation

Dependency	appearance.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-appearance
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

6.2.1. Dependencies

The Appearance Requirements Classes is dependent on the following Requirements Classes from this standard.

- The Base Requirements Class is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Base Requirements Class also defines XML elements and types reused by other Requirements Classes.
- GML 3.2 provides most of the geometry types (e.g., Point, LineString, Polygon) used for spatial representations in this Standard. Defines Coordinate Reference Systems. Supports the General Feature Model upon which this Standard is based.
- GML 3.3 defines the linear referencing concepts (e.g., linear element, distance along, Linear Referencing Methods) used for linearly referenced locations in this Standard. Also included are TINs.

6.2.2. GML Elements

The CityGML Appearance XML elements and their corresponding CityGML UML classes are shown in Table 4.

Table 4. Appearance XML Elements

XML Schema Element	Conceptual Model
AbstractSurfaceData	«FeatureType» AbstractSurfaceData
AbstractTexture	«FeatureType» AbstractTexture
Appearance	«FeatureType» Appearance
GeoreferencedTexture	«FeatureType» GeoreferencedTexture
ParameterizedTexture	«FeatureType» ParameterizedTexture
TextureAssociation	«ObjectType» TextureAssociation
X3DMaterial	«FeatureType» X3DMaterial
AbstractTextureParameterization	«DataType» AbstractTextureParameterization
ADEOfAbstractSurfaceData	«DataType» ADEOfAbstractSurfaceData
ADEOfAbstractTexture	«DataType» ADEOfAbstractTexture
ADEOfAppearance	«DataType» ADEOfAppearance
ADEOfGeoreferencedTexture	«DataType» ADEOfGeoreferencedTexture
ADEOfParameterizedTexture	«DataType» ADEOfParameterizedTexture
ADEOfX3DMaterial	«DataType» ADEOfX3DMaterial

XML Schema Element	Conceptual Model
TexCoordGen	«DataType» TexCoordGen
TexCoordList	«DataType» TexCoordList

6.2.3. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 `Appearance` conformance class in GML.

1. decision 1
2. decision 2

6.2.4. Requirements

Requirement 2	/req/base/elements
A conforming application shall support the CityGML XML elements listed in Table 4 in accordance with the GML XML schema specified in appearance.xsd.	

6.3. Bridge Conformance Class

The Bridge Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Bridge: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-bridge

The applicable GML schema is bridge.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-bridgee	
Target type	Implementation
Dependency	bridge.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-bridge
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 3	/req/bridge/elements
A conforming application shall support the CityGML XML elements listed in [bridge-xml-elements] in accordance with the GML XML schema specified in bridge.xsd.	

6.4. Building Conformance Class

The Building Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Building: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-building

The applicable GML schema is building.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-building	
Target type	Implementation
Dependency	building.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-building
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 4	**/req/building/elements**
A conforming application shall support the CityGML XML elements listed in [building-xml-elements] in accordance with the GML XML schema specified in bulding.xsd.	

6.5. City Furniture Conformance Class

The City Furniture Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- CityFurniture: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityfurniture

The applicable GML schema is cityFurniture.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-cityfurniture	
Target type	Implementation
Dependency	cityFurniture.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityfurniture
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 5	**/req/base/cityfurniture**
A conforming application shall support the CityGML XML elements listed in [city-furniture-xml-elements] in accordance with the GML XML schema specified in cityFurniture.xsd.	

6.6. City Object Group Conformance Class

The City Object Group Conformance Class implements the following Requirement Classe from the CityGML 3.0 Conceptual Model Standard:

- CityObjectGroup: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityobjectgroup

The applicable GML schema is cityObjectGroup.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-cityobjectgroup	
Target type	Implementation
Dependency	cityObjectGroup.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityobjectgroup
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 6	/req/cityobjectgroup/elements
A conforming application shall support the CityGML XML elements listed in [city-object-group-xml-elements] in accordance with the GML XML schema specified in cityObjectGroup.xsd.	

6.7. Construction Conformance Class

The Construction Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Construction: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-construction

The applicable GML schema is construction.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-construction	
Target type	Implementation
Dependency	construction.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-construction
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 7	/req/construction/elements
A conforming application shall support the CityGML XML elements listed in [construction-xml-elements] in accordance with the GML XML schema specified in construction.xsd.	

6.8. Dynamizer Conformance Class

The Dynamizer Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Dynamizer: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-dynamizer

The applicable GML schema is dynamizer.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-dynamizer	
Target type	Implementation
Dependency	dynamizer.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-dynamizer
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 8	/req/dynamizer/elements
A conforming application shall support the CityGML XML elements listed in [dynamizer-xml-elements] in accordance with the GML XML schema specified in dynamizer.xsd.	

6.9. Generics Conformance Class

The Generics Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Generics: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-generics

The applicable GML schema is generics.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-generics	
Target type	Implementation
Dependency	generics.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-generics
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 9	/req/generics/elements
A conforming application shall support the CityGML XML elements listed in [generics-xml-elements] in accordance with the GML XML schema specified in generics.xsd.	

6.10. Land Use Conformance Class

The Land Use Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- LandUse: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-landuse

The applicable GML schema is landUse.xsd

Requirements Class

http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-landuse	
Target type	Implementation
Dependency	landUse.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-landuse
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 10	**/req/landuse/elements**
A conforming application shall support the CityGML XML elements listed in [land-use-xml-elements] in accordance with the GML XML schema specified in landUse.xsd.	

6.11. Point Cloud Conformance Class

The Point Cloud Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Point Cloud: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud

The applicable GML schema is pointCloud.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-pointcloud	
Target type	Implementation
Dependency	pointCloud.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 11	**/req/pointcloud/elements**
A conforming application shall support the CityGML XML elements listed in [point-cloud-xml-elements] in accordance with the GML XML schema specified in pointCloud.xsd.	

6.12. Relief Conformance Class

The Relief Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Relief: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief

The applicable GML schema is relief.xsd

Requirements Class
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-relief

Target type	Implementation
Dependency	relief.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 12	**/req/relief/elements**
A conforming application shall support the CityGML XML elements listed in [relief-xml-elements] in accordance with the GML XML schema specified in relief.xsd.	

6.13. Transportation Conformance Class

The Transportation Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Transportation: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-transportation

The applicable GML schema is transportation.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-transportation	
Target type	Implementation
Dependency	transportation.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-transportation
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 13	**/req/base/transportation**
A conforming application shall support the CityGML XML elements listed in [transportation-xml-elements] in accordance with the GML XML schema specified in transportation.xsd.	

6.14. Tunnel Conformance Class

The Tunnel Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Tunnel: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel

The applicable GML schema is tunnel.xsd

Requirements Class
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-tunnel

Target type	Implementation
Dependency	tunnel.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 14	**/req/base/tunnel**
A conforming application shall support the CityGML XML elements listed in [tunnel-xml-elements] in accordance with the GML XML schema specified in tunnel.xsd.	

6.15. Vegetation Conformance Class

The Vegetation Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Vegetation: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-vegetation

The applicable GML schema is vegetation.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-vegetation	
Target type	Implementation
Dependency	vegetation.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-appearance
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-vegetation

Requirement 15	**/req/vegetation/elements**
A conforming application shall support the CityGML XML elements listed in [vegetation-xml-elements] in accordance with the GML XML schema specified in vegetation.xsd.	

6.16. Versioning Conformance Class

The Versioning Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- Versioning: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-versioning

The applicable GML schema is versioning.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-versioning	
Target type	Implementation
Dependency	versioning.xsd

Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-versioning
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 16	**/req/versioning/elements**
A conforming application shall support the CityGML XML elements listed in [verisioning-xml-elements] in accordance with the GML XML schema specified in versioning.xsd.	

6.17. Water Body Conformance Class

The Water Body Conformance Class implements the following Requirements Classes from the CityGML 3.0 Conceptual Model Standard:

- WaterBody: http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody

The applicable GML schema is waterBody.xsd

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-waterbody	
Target type	Implementation
Dependency	waterBody.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-base

Requirement 17	**/req/waterbody/elements**
A conforming application shall support the CityGML XML elements listed in [water-body-xml-elements] in accordance with the GML XML schema specified in waterBody.xsd.	

Chapter 7. Media Types

Data for all Parts of the CityGML 3.0 GML encoding standard is encoded in GML-conformant XML documents. The standard MIME-type and sub-type for GML data should be used to indicate the encoding in internet exchange.

The registered MIME Media Type for GML documents is `application/gml+xml`.

Annex A: Conformance Class Abstract Test Suite (Normative)

NOTE	Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

A.1. Conformance Class A

A.1.1. Requirement 1

Test id:	/conf/conf-class-a/req-name-1
Requirement:	/req/req-class-a/req-name-1
Test purpose:	Verify that…
Test method:	Inspect…

A.1.2. Requirement 2

Annex B: Examples (Informative)

NOTE	This is where any examples will live. For ease of maintenance, each example should be created in its' own asccidoc file and then imported using an "include" statement.

Annex C: Schema (Normative)

NOTE	This is where any XML or JSON schema reside. Conformance is defined, in part, by conformance to these schema.

Annex D: Conceptual Model Conformance (Normative)

NOTE	This is where conformance with CityGML 3.0 Conceptual Model is documented.

Annex E: CityGML Data Dictionary (Normative)

The CityGML UML model is the normative definition of the CityGML Conceptual Model. The Data Dictionary tables in this section were software generated from the UML model. As such, this section provides a normative representation of the CityGML Conceptual Model.

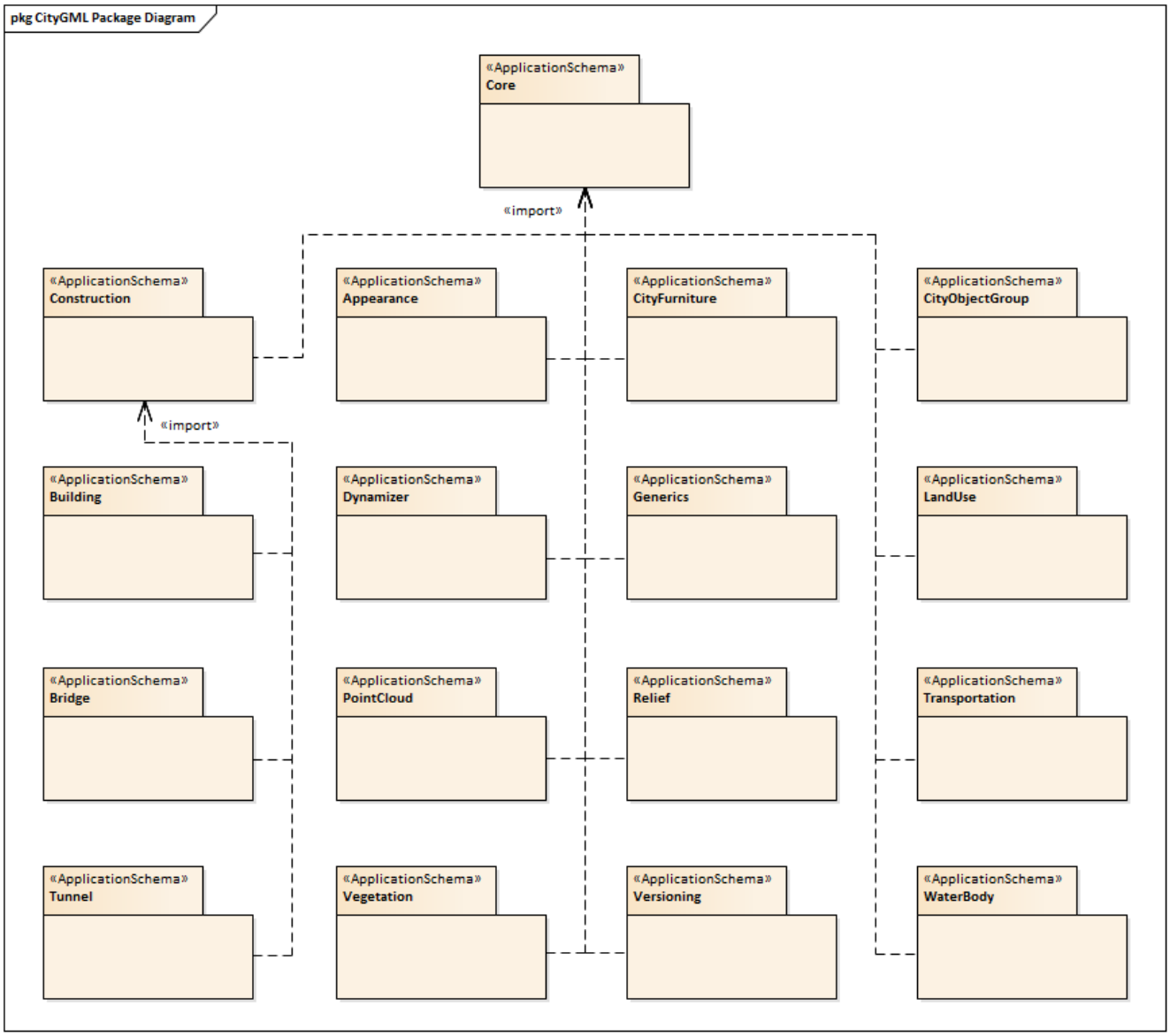


Figure 3. CityGML UML Packages

E.1. Core

Description:	The Core module defines the basic components of the CityGML data model. The Core module defines abstract base classes that define the core properties of more specialized thematic classes defined in other modules. The Core module also defines concrete classes that are common to other modules, for example basic data types.
Parent Package:	CityGML

Stereotype: «ApplicationSchema»

E.1.1. Classes

AbstractAppearance

Definition: AbstractAppearance is the abstract superclass to represent any kind of appearance objects.

Subclass of: AbstractFeatureWithLifespan

Stereotype: «FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractAppearance	ADEOfAbstractAppearance [0..*]	Augments AbstractAppearance with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractCityObject

Definition: AbstractCityObject is the abstract superclass of all thematic classes within the CityGML Conceptual Model.

Subclass of: AbstractFeatureWithLifespan

Stereotype: «FeatureType»

Role name	Target class and multiplicity	Definition
generalizesTo	AbstractCityObject [*]	Relates generalized representations of the same real-world object in different Levels of Detail to the city object. The direction of this relation is from the city object to the corresponding generalized city objects.
genericAttribute	AbstractGenericAttribute [*]	Relates generic attributes to the city object.
dynamizer	AbstractDynamizer [*]	Relates Dynamizer objects to the city object. These allow timeseries data to override static attribute values of the city object.
appearance	AbstractAppearance [*]	Relates appearances to the city object.
externalReference	ExternalReference [*]	References external objects in other information systems that have a relation to the city object.
relatedTo	AbstractCityObject [*]	Relates other city objects to the city object. It also describes how the city objects are related to each other.

Attribute	Value type and multiplicity	Definition
relativeToTerrain	RelativeToTerrain [0..1]	Describes the vertical position of the city object relative to the surrounding terrain.
relativeToWater	RelativeToWater [0..1]	Describes the vertical position of the city object relative to the surrounding water surface.
adeOfAbstractCityObject	ADEOfAbstractCityObject [0..*]	Augments AbstractCityObject with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractDynamizer

Definition:	AbstractDynamizer is the abstract superclass to represent Dynamizer objects.
Subclass of:	AbstractFeatureWithLifespan
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractDynamizer	ADEOfAbstractDynamizer [0..*]	Augments AbstractDynamizer with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractFeature

Definition:	AbstractFeature is the abstract superclass of all feature types within the CityGML Conceptual Model.
Subclass of:	AnyFeature
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
featureID	ID [1..1]	Specifies the unique identifier of the feature that is valid in the instance document within which it occurs.
identifier	ScopedName [0..1]	Specifies the unique identifier of the feature that is valid globally.
name	GenericName [0..*]	Specifies the name of the feature.
description	CharacterString [0..1]	Provides further information on the feature.
adeOfAbstractFeature	ADEOfAbstractFeature [0..*]	Augments AbstractFeature with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractFeatureWithLifespan

Definition:	AbstractFeatureWithLifespan is the base class for all CityGML features. This class allows the optional specification of the real-world and database times for the existence of each feature.
Subclass of:	AbstractFeature
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
creationDate	DateTime [0..1]	Indicates the date at which a CityGML feature was added to the CityModel.
terminationDate	DateTime [0..1]	Indicates the date at which a CityGML feature was removed from the CityModel.
validFrom	DateTime [0..1]	Indicates the date at which a CityGML feature started to exist in the real world.
validTo	DateTime [0..1]	Indicates the date at which a CityGML feature ended to exist in the real world.
adeOfAbstractFeatureWithLifespan	ADEOfAbstractFeatureWithLifespan [0..*]	Augments AbstractFeatureWithLifespan with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractLogicalSpace

Definition:	AbstractLogicalSpace is the abstract superclass for all types of logical spaces. Logical space refers to spaces that are not bounded by physical surfaces but are defined according to thematic considerations.
Subclass of:	AbstractSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractLogicalSpace	ADEOfAbstractLogicalSpace [0..*]	Augments AbstractLogicalSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractOccupiedSpace

Definition:	AbstractOccupiedSpace is the abstract superclass for all types of physically occupied spaces. Occupied space refers to spaces that are partially or entirely filled with matter.
Subclass of:	AbstractPhysicalSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
lod3ImplicitRepresentation	ImplicitGeometry [0..1]	Relates to an implicit geometry that represents the occupied space in Level of Detail 3.
lod1ImplicitRepresentation	ImplicitGeometry [0..1]	Relates to an implicit geometry that represents the occupied space in Level of Detail 1.
lod2ImplicitRepresentation	ImplicitGeometry [0..1]	Relates to an implicit geometry that represents the occupied space in Level of Detail 2.

Attribute	Value type and multiplicity	Definition
adeOfAbstractOccupiedSpace	ADEOfAbstractOccupiedSpace [0..*]	Augments AbstractOccupiedSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractPhysicalSpace

Definition:	AbstractPhysicalSpace is the abstract superclass for all types of physical spaces. Physical space refers to spaces that are fully or partially bounded by physical objects.
Subclass of:	AbstractSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
lod3TerrainIntersectionCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the terrain intersection curve of the physical space in Level of Detail 3.
lod2TerrainIntersectionCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the terrain intersection curve of the physical space in Level of Detail 2.
pointCloud	AbstractPointCloud [0..1]	Relates to a 3D PointCloud that represents the physical space.
lod1TerrainIntersectionCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the terrain intersection curve of the physical space in Level of Detail 1.

Attribute	Value type and multiplicity	Definition
adeOfAbstractPhysicalSpace	ADEOfAbstractPhysicalSpace [0..*]	Augments AbstractPhysicalSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractPointCloud

Definition:	AbstractPointCloud is the abstract superclass to represent PointCloud objects.
Subclass of:	AbstractFeature
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractPointCloud	ADEOfAbstractPointCloud [0..*]	Augments AbstractPointCloud with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractSpace

Definition:	AbstractSpace is the abstract superclass for all types of spaces. A space is an entity of volumetric extent in the real world.
Subclass of:	AbstractCityObject
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
lod2MultiCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the space in Level of Detail 2.
lod0MultiCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the space in Level of Detail 0.
lod0MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the space in Level of Detail 0.
lod2MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the space in Level of Detail 2.
lod3MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the space in Level of Detail 3.
lod0Point	GM_Point [0..1]	Relates to a 3D Point geometry that represents the space in Level of Detail 0.
lod3Solid	GM_Solid [0..1]	Relates to a 3D Solid geometry that represents the space in Level of Detail 3.
lod3MultiCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the space in Level of Detail 3.
lod2Solid	GM_Solid [0..1]	Relates to a 3D Solid geometry that represents the space in Level of Detail 2.
boundary	AbstractSpaceBoundary [*]	Relates to surfaces that bound the space.
lod1Solid	GM_Solid [0..1]	Relates to a 3D Solid geometry that represents the space in Level of Detail 1.

Attribute	Value type and multiplicity	Definition
spaceType	SpaceType [0..1]	Specifies the degree of openness of a space.
volume	QualifiedVolume [0..*]	Specifies qualified volumes related to the space.
area	QualifiedArea [0..*]	Specifies qualified areas related to the space.
adeOfAbstractSpace	ADEOfAbstractSpace [0..*]	Augments AbstractSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractSpaceBoundary

Definition:	AbstractSpaceBoundary is the abstract superclass for all types of space boundaries. A space boundary is an entity with areal extent in the real world. Space boundaries are objects that bound a Space. They also realize the contact between adjacent spaces.
Subclass of:	AbstractCityObject
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractSpaceBoundary	ADEOfAbstractSpaceBoundary [0..*]	Augments AbstractSpaceBoundary with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractThematicSurface

Definition:	AbstractThematicSurface is the abstract superclass for all types of thematic surfaces.
Subclass of:	AbstractSpaceBoundary
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
lod1MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 1.
pointCloud	AbstractPointCloud [0..1]	Relates to a 3D PointCloud that represents the thematic surface.
lod0MultiCurve	GM_MultiCurve [0..1]	Relates to a 3D MultiCurve geometry that represents the thematic surface in Level of Detail 0.
lod3MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 3.
lod0MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 0.
lod2MultiSurface	GM_MultiSurface [0..1]	Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 2.

Attribute	Value type and multiplicity	Definition
area	QualifiedArea [0..*]	Specifies qualified areas related to the thematic surface.
adeOfAbstractThematicSurface	ADEOfAbstractThematicSurface [0..*]	Augments AbstractThematicSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractUnoccupiedSpace

Definition:	AbstractUnoccupiedSpace is the abstract superclass for all types of physically unoccupied spaces. Unoccupied space refers to spaces that are entirely or mostly free of matter.
Subclass of:	AbstractPhysicalSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractUnoccupiedSpace	ADEOfAbstractUnoccupiedSpace [0..*]	Augments AbstractUnoccupiedSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractVersion

Definition:	AbstractVersion is the abstract superclass to represent Version objects.
Subclass of:	AbstractFeatureWithLifespan
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractVersion	ADEOfAbstractVersion [0..*]	Augments AbstractVersion with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractVersionTransition

Definition:	AbstractVersionTransition is the abstract superclass to represent VersionTransition objects.
Subclass of:	AbstractFeatureWithLifespan
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractVersionTransition	ADEOfAbstractVersionTransition [0..*]	Augments AbstractVersionTransition with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Address

Definition:	Address represents an address of a city object.
Subclass of:	AbstractFeature
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
multiPoint	GM_MultiPoint [0..1]	Relates to the MultiPoint geometry of the Address. The geometry relates the address spatially to a city object.
xalAddress	XALAddress [1..1]	Relates an OASIS address object to the Address.

Attribute	Value type and multiplicity	Definition
adeOfAddress	ADEOfAddress [0..*]	Augments the Address with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

CityModel

Definition:	CityModel is the container for all objects belonging to a city model.
Subclass of:	AbstractFeatureWithLifespan
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
cityModelMember	CityModelMember [*]	Relates to all objects that are part of the CityModel.

Attribute	Value type and multiplicity	Definition
engineeringCRS	EngineeringCRS [0..1]	Specifies the local engineering coordinate reference system of the CityModel that can be provided inline the CityModel instead of referencing a well-known CRS definition. The definition of an engineering CRS requires an anchor point which relates the origin of the local coordinate system to a point on the earth's surface in order to facilitate the transformation of coordinates from the local engineering CRS.
adeOfCityModel	ADEOfCityModel [0..*]	Augments the CityModel with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

CityObjectRelation

Definition:	CityObjectRelation represents a specific relation from the city object in which the relation is included to another city object.
Subclass of:	None
Stereotype:	«ObjectType»

Role name	Target class and multiplicity	Definition
genericAttribute	AbstractGenericAttribute [*]	Relates generic attributes to the CityObjectRelation.

Attribute	Value type and multiplicity	Definition
relationType	RelationTypeValue [1..1]	Indicates the specific type of the CityObjectRelation.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

ClosureSurface

Definition:	ClosureSurface is a special type of thematic surface used to close holes in volumetric objects. Closure surfaces are virtual (non-physical) surfaces.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfClosureSurface	ADEOfClosureSurface [0..*]	Augments the ClosureSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

ImplicitGeometry

Definition:	ImplicitGeometry is a geometry representation where the shape is stored only once as a prototypical geometry. For example a tree or other vegetation object, a traffic light or a traffic sign. This prototypic geometry object can be re-used or referenced many times, wherever the corresponding feature occurs in the 3D city model.
Subclass of:	None
Stereotype:	«ObjectType»

Role name	Target class and multiplicity	Definition
relativeGeometry	GM_Object [0..1]	Relates to a prototypical geometry in a local coordinate system stored inline with the city model.
referencePoint	GM_Point [1..1]	Relates to a 3D Point geometry that represents the base point of the object in the world coordinate system.
appearance	AbstractAppearance [*]	Relates appearances to the ImplicitGeometry.

Attribute	Value type and multiplicity	Definition
objectID	ID [1..1]	Specifies the unique identifier of the ImplicitGeometry.
transformationMatrix	TransformationMatrix4x4 [1..1]	Specifies the mathematical transformation (translation, rotation, and scaling) between the prototypical geometry and the actual spatial position of the object.
mimeType	MimeTypeValue [0..1]	Specifies the MIME type of the external file that stores the prototypical geometry.
libraryObject	URI [0..1]	Specifies the URI that points to the prototypical geometry stored in an external file.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.1.2. Data Types

AbstractGenericAttribute

Definition:	AbstractGenericAttribute is the abstract superclass for all types of generic attributes.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractAppearance

Definition:	ADEOfAbstractAppearance acts as a hook to define properties within an ADE that are to be added to AbstractAppearance.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractCityObject

Definition:	ADEOfAbstractCityObject acts as a hook to define properties within an ADE that are to be added to AbstractCityObject.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractDynamizer

Definition:	ADEOfAbstractDynamizer acts as a hook to define properties within an ADE that are to be added to AbstractDynamizer.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractFeature

Definition:	ADEOfAbstractFeature acts as a hook to define properties within an ADE that are to be added to AbstractFeature.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractFeatureWithLifespan

Definition:	ADEOfAbstractFeatureWithLifespan acts as a hook to define properties within an ADE that are to be added to AbstractFeatureWithLifespan.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractLogicalSpace

Definition:	ADEOfAbstractLogicalSpace acts as a hook to define properties within an ADE that are to be added to AbstractLogicalSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractOccupiedSpace

Definition:	ADEOfAbstractOccupiedSpace acts as a hook to define properties within an ADE that are to be added to AbstractOccupiedSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractPhysicalSpace

Definition:	ADEOfAbstractPhysicalSpace acts as a hook to define properties within an ADE that are to be added to AbstractPhysicalSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractPointCloud

Definition:	ADEOfAbstractPointCloud acts as a hook to define properties within an ADE that are to be added to AbstractPointCloud.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractSpace

Definition:	ADEOfAbstractSpace acts as a hook to define properties within an ADE that are to be added to AbstractSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractSpaceBoundary

Definition:	ADEOfAbstractSpaceBoundary acts as a hook to define properties within an ADE that are to be added to AbstractSpaceBoundary.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractThematicSurface

Definition:	ADEOfAbstractThematicSurface acts as a hook to define properties within an ADE that are to be added to AbstractThematicSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractUnoccupiedSpace

Definition:	ADEOfAbstractUnoccupiedSpace acts as a hook to define properties within an ADE that are to be added to AbstractUnoccupiedSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractVersion

Definition:	ADEOfAbstractVersion acts as a hook to define properties within an ADE that are to be added to AbstractVersion.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractVersionTransition

Definition:	ADEOfAbstractVersionTransition acts as a hook to define properties within an ADE that are to be added to AbstractVersionTransition.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAddress

Definition:	ADEOfAddress acts as a hook to define properties within an ADE that are to be added to an Address.
Subclass of:	None
Stereotype:	«DataType»

ADEOfCityModel

Definition:	ADEOfCityModel acts as a hook to define properties within an ADE that are to be added to a CityModel.
Subclass of:	None
Stereotype:	«DataType»

ADEOfClosureSurface

Definition:	ADEOfClosureSurface acts as a hook to define properties within an ADE that are to be added to a ClosureSurface.
Subclass of:	None
Stereotype:	«DataType»

ExternalReference

Definition:	ExternalReference is a reference to a corresponding object in another information system, for example in the German cadastre (ALKIS), the German topographic information system (ATKIS), or the OS UK MasterMap®.
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
targetResource	URI [1..1]	Specifies the URI that points to the object in the external information system.
informationSystem	URI [0..1]	Specifies the URI that points to the external information system.
relationType	URI [0..1]	Specifies a URI that additionally qualifies the ExternalReference. The URI can point to a definition from an external ontology (e.g. the sameAs relation from OWL) and allows for mapping the ExternalReference to RDF triples.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Occupancy

Definition:	Occupancy is an application-dependent indication of what is contained by a feature.
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
numberOfOccupants	Integer [1..1]	Indicates the number of occupants contained by a feature.
interval	IntervalValue [0..1]	Indicates the time period the occupants are contained by a feature.
occupantType	OccupantTypeValue [0..1]	Indicates the specific type of the occupants that are contained by a feature.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

QualifiedArea

Definition:	QualifiedArea is an application-dependent measure of the area of a space or of a thematic surface.
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
area	Area [1..1]	Specifies the value of the QualifiedArea.
typeOfArea	QualifiedAreaTypeValue [1..1]	Indicates the specific type of the QualifiedArea.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

QualifiedVolume

Definition:	QualifiedVolume is an application-dependent measure of the volume of a space.
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
volume	Volume [1..1]	Specifies the value of the QualifiedVolume.
typeOfVolume	QualifiedVolumeTypeValue [1..1]	Indicates the specific type of the QualifiedVolume.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

XALAddress

Definition:	XALAddress represents address details according to the OASIS xAL standard.
Subclass of:	None
Stereotype:	«DataType»

E.1.3. Basic Types

Code

Definition:	Code is a basic type for a String-based term, keyword, or name that can additionally have a code space.
Subclass of:	None
Stereotype:	«BasicType»

Attribute	Value type and multiplicity	Definition
codeSpace	URI [0..1]	Associates the Code with an authority that controls the term, keyword, or name.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

DoubleBetween0and1

Definition:	DoubleBetween0and1 is a basic type for values, which are greater or equal than 0 and less or equal than 1. The type is used for color encoding, for example.
Subclass of:	None
Stereotype:	«BasicType»
Constraint:	valueBetween0and1 (OCL): inv: DoubleBetween0and1.allInstances() → forAll(p | p > = 0 and p < = 1)

DoubleBetween0and1List

Definition:	DoubleBetween0and1List is a basic type that represents a list of double values greater or equal than 0 and less or equal than 1. The type is used for color encoding, for example.
Subclass of:	None
Stereotype:	«BasicType»

Attribute	Value type and multiplicity	Definition
list	DoubleBetween0and1 [1..1]	Specifies the list of double values.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

DoubleList

Definition:	DoubleList is an ordered sequence of double values.
Subclass of:	None
Stereotype:	«BasicType»

Attribute	Value type and multiplicity	Definition
list	Real [1..1]	Specifies the list of double values.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

DoubleOrNilReasonList

Definition:	DoubleOrNilReasonList is a basic type that represents a list of double values and/or nil reasons.
Subclass of:	None
Stereotype:	«BasicType»

Attribute	Value type and multiplicity	Definition
list	DoubleOrNilReason [1..1]	Specifies the list of double values and/or nil reasons.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

ID

Definition:	ID is a basic type that represents a unique identifier.
Subclass of:	None
Stereotype:	«BasicType»

IntegerBetween0and3

Definition:	IntegerBetween0and3 is a basic type for integer values, which are greater or equal than 0 and less or equal than 3. The type is used for encoding the LOD number.
Subclass of:	None
Stereotype:	«BasicType»
Constraint:	valueBetween0and3 (OCL): inv: IntegerBetween0and3.allInstances() → forAll(p | p > = 0 and p < = 3)

MeasureOrNilReasonList

Definition:	MeasureOrNilReasonList is a basic type that represents a list of double values and/or nil reasons together with a unit of measurement.
Subclass of:	DoubleOrNilReasonList
Stereotype:	«BasicType»

Attribute	Value type and multiplicity	Definition
uom	UnitOfMeasure [1..1]	Specifies the unit of measurement of the double values.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TransformationMatrix2x2

Definition:	TransformationMatrix2x2 is a 2 by 2 matrix represented as a list of four double values in row major order.
Subclass of:	DoubleList
Stereotype:	«BasicType»
Constraint:	lengthOfList (OCL): inv: list→size() = 4

TransformationMatrix3x4

Definition:	TransformationMatrix3x4 is a 3 by 4 matrix represented as a list of twelve double values in row major order.
Subclass of:	DoubleList
Stereotype:	«BasicType»
Constraint:	lengthOfList (OCL): inv: list→size() = 12

TransformationMatrix4x4

Definition:	TransformationMatrix4x4 is a 4 by 4 matrix represented as a list of sixteen double values in row major order.
Subclass of:	DoubleList
Stereotype:	«BasicType»
Constraint:	lengthOfList (OCL): inv: list → size() = 16

E.1.4. Unions

CityModelMember

Definition:	CityModelMember is a union type that enumerates the different types of objects that can occur as members of a city model.
Stereotype:	«Union»

Member name	Type	Definition
cityObjectMember	AbstractCityObject [1..1]	Specifies the city objects that are part of the CityModel.
appearanceMember	AbstractAppearance [1..1]	Specifies the appearances of the CityModel.
versionMember	AbstractVersion [1..1]	Specifies the different versions of the CityModel.
versionTransitionMember	AbstractVersionTransition [1..1]	Specifies the transitions between the different versions of the CityModel.
featureMember	AbstractFeature [1..1]	Specifies the feature objects that are part of the CityModel. It allows to include objects that are not derived from a class defined in the CityGML conceptual model, but from the ISO 19109 class AnyFeature.

DoubleOrNilReason

Definition:	DoubleOrNilReason is a union type that allows for choosing between a double value and a nil reason.
Stereotype:	«Union»

Member name	Type	Definition
value	Real [1..1]	Specifies the double value.
nilReason	NilReason [1..1]	Specifies the nil reason.

NilReason

Definition:	NilReason is a union type that allows for choosing between two different types of nil reason.
Stereotype:	«Union»

Member name	Type	Definition
nilReasonEnumeration	NilReasonEnumeration [1..1]	Indicates a nil reason that is provided in a code list.
URI	URI [1..1]	Specifies a URI that points to a resource that describes the nil reason.

E.1.5. Code Lists

IntervalValue

Definition:	IntervalValue is a code list used to specify a time period.
Stereotype:	«CodeList»

MimeTypeValue

Definition:	MimeTypeValue is a code list used to specify the MIME type of a referenced resource.
Stereotype:	«CodeList»

NilReasonEnumeration

Definition:	NilReasonEnumeration is a code list that enumerates the different nil reasons.
Stereotype:	«CodeList»

OccupantTypeValue

Definition:	OccupantTypeValue is a code list used to classify occupants.
Stereotype:	«CodeList»

OtherRelationTypeValue

Definition:	OtherRelationTypeValue is a code list used to classify other types of city object relations.
Stereotype:	«CodeList»

QualifiedAreaTypeValue

Definition:	QualifiedAreaTypeValue is a code list used to specify area types.
Stereotype:	«CodeList»

QualifiedVolumeTypeValue

Definition:	QualifiedVolumeTypeValue is a code list used to specify volume types.
Stereotype:	«CodeList»

RelationTypeValue

Definition:	RelationTypeValue is a code list used to classify city object relations.
Stereotype:	«CodeList»

TemporalRelationTypeValue

Definition:	TemporalRelationTypeValue is a code list used to classify temporal city object relations.
Stereotype:	«CodeList»

TopologicalRelationTypeValue	
Definition:	TopologicalRelationTypeValue is a code list used to classify topological city object relations.
Stereotype:	«CodeList»

E.1.6. Enumerations

RelativeToTerrain	
Definition:	RelativeToTerrain enumerates the spatial relations of a city object relative to terrain in a qualitative way.
StereoType:	<<Enumeration>>

Literal value	Definition
entirelyAboveTerrain	Indicates that the city object is located entirely above the terrain.
substantiallyAboveTerrain	Indicates that the city object is for the most part located above the terrain.
substantiallyAboveAndBelowTerrain	Indicates that the city object is located half above the terrain and half below the terrain.
substantiallyBelowTerrain	Indicates that the city object is for the most part located below the terrain.
entirelyBelowTerrain	Indicates that the city object is located entirely below the terrain.

RelativeToWater	
Definition:	RelativeToWater enumerates the spatial relations of a city object relative to the water surface in a qualitative way.
StereoType:	<<Enumeration>>

Literal value	Definition
entirelyAboveWaterSurface	Indicates that the city object is located entirely above the water surface.
substantiallyAboveWaterSurface	Indicates that the city object is for the most part located above the water surface.
substantiallyAboveAndBelowWaterSurface	Indicates that the city object is located half above the water surface and half below the water surface.
substantiallyBelowWaterSurface	Indicates that the city object is for the most part located below the water surface.
entirelyBelowWaterSurface	Indicates that the city object is located entirely below the water surface.
temporarilyAboveAndBelowWaterSurface	Indicates that the city object is temporarily located above or below the water level, because the height of the water surface is varying.

SpaceType

Definition:	SpaceType is an enumeration that characterises a space according to its closure properties.
StereoType:	<<Enumeration>>

Literal value	Definition
closed	Indicates that the space has boundaries at the bottom, at the top, and on all sides.
open	Indicates that the space has at maximum a boundary at the bottom.
semiOpen	Indicates that the space has a boundary at the bottom and on at least one side.

E.2. Appearance

Description:	The Appearance module supports the modelling of the observable surface properties of CityGML features in the form of textures and material.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.2.1. Classes

AbstractSurfaceData

Definition:	AbstractSurfaceData is the abstract superclass for different kinds of textures and material.
Subclass of:	AbstractFeature
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
isFront	Boolean [0..1]	Indicates whether the texture or material is assigned to the front side or the back side of the surface geometry object.
adeOfAbstractSurfaceData	ADEOfAbstractSurfaceData [0..*]	Augments AbstractSurfaceData with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractTexture

Definition:	AbstractTexture is the abstract superclass to represent the common attributes of the classes ParameterizedTexture and GeoreferencedTexture.
Subclass of:	AbstractSurfaceData
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
imageURI	URI [1..1]	Specifies the URI that points to the external image data file.
mimeType	MimeTypeValue [0..1]	Specifies the MIME type of the external point cloud file.
textureType	TextureType [0..1]	Indicates the specific type of the texture.
wrapMode	WrapMode [0..1]	Specifies the behaviour of the texture when the texture is smaller than the surface to which it is applied.
borderColor	ColorPlusOpacity [0..1]	Specifies the color of that part of the surface that is not covered by the texture.
adeOfAbstractTexture	ADEOfAbstractTexture [0..*]	Augments AbstractTexture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Appearance

Definition:	An Appearance is a collection of surface data, i.e. observable properties for surface geometry objects in the form of textures and material.
Subclass of:	AbstractAppearance
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
surfaceData	AbstractSurfaceData [*]	Relates to the surface data that are part of the Appearance.

Attribute	Value type and multiplicity	Definition
theme	CharacterString [0..1]	Specifies the topic of the Appearance. Each Appearance contains surface data for one theme only. Examples of themes are infrared radiation, noise pollution, or earthquake-induced structural stress.
adeOfAppearance	ADEOfAppearance [0..*]	Augments the Appearance with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GeoreferencedTexture

Definition:	A GeoreferencedTexture is a texture that uses a planimetric projection. It contains an implicit parameterization that is either stored within the image file, an accompanying world file or specified using the orientation and referencePoint elements.
Subclass of:	AbstractTexture
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
referencePoint	GM_Point [0..1]	Relates to the 2D Point geometry that represents the center of the upper left image pixel in world space.

Attribute	Value type and multiplicity	Definition
preferWorldFile	Boolean [0..1]	Indicates whether the georeference from the image file or the accompanying world file should be preferred.
orientation	TransformationMatrix2x2 [0..1]	Specifies the rotation and scaling of the image in form of a 2x2 matrix.
target	URI [0..*]	Specifies the URI that points to the surface geometry objects to which the texture is applied.
adeOfGeoreferencedTexture	ADEOfGeoreferencedTexture [0..*]	Augments the GeoreferencedTexture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

ParameterizedTexture

Definition:	A ParameterizedTexture is a texture that uses texture coordinates or a transformation matrix for parameterization.
Subclass of:	AbstractTexture
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
textureParameterization	AbstractTextureParameterization [*]	Relates to the texture coordinates or transformation matrices used for parameterization.

Attribute	Value type and multiplicity	Definition
adeOfParameterizedTexture	ADEOfParameterizedTexture [0..*]	Augments the ParameterizedTexture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TextureAssociation

Definition:	TextureAssociation denotes the relation of a texture to a surface geometry object.
Subclass of:	None
Stereotype:	«ObjectType»

Attribute	Value type and multiplicity	Definition
target	URI [1..1]	Specifies the URI that points to the surface geometry object to which the texture is applied.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

X3DMaterial

Definition:	X3DMaterial defines properties for surface geometry objects based on the material definitions from the X3D and COLLADA standards.
Subclass of:	AbstractSurfaceData
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
ambientIntensity	DoubleBetween0and1 [0..1]	Specifies the minimum percentage of diffuseColor that is visible regardless of light sources.
diffuseColor	Color [0..1]	Specifies the color of the light diffusely reflected by the surface geometry object.
emissiveColor	Color [0..1]	Specifies the color of the light emitted by the surface geometry object.
specularColor	Color [0..1]	Specifies the color of the light directly reflected by the surface geometry object.
shininess	DoubleBetween0and1 [0..1]	Specifies the sharpness of the specular highlight.
transparency	DoubleBetween0and1 [0..1]	Specifies the degree of transparency of the surface geometry object.
isSmooth	Boolean [0..1]	Specifies which interpolation method is used for the shading of the surface geometry object. If the attribute is set to true, vertex normals should be used for shading (Gouraud shading). Otherwise, normals should be constant for a surface patch (flat shading).
target	URI [0..*]	Specifies the URI that points to the surface geometry objects to which the material is applied.
adeOfX3DMaterial	ADEOfX3DMaterial [0..*]	Augments the X3DMaterial with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.2.2. Data Types

AbstractTextureParameterization

Definition:	AbstractTextureParameterization is the abstract superclass for different kinds of texture parameterizations.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractSurfaceData

Definition:	ADEOfAbstractSurfaceData acts as a hook to define properties within an ADE that are to be added to AbstractSurfaceData.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractTexture

Definition:	ADEOfAbstractTexture acts as a hook to define properties within an ADE that are to be added to AbstractTexture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAppearance

Definition:	ADEOfAppearance acts as a hook to define properties within an ADE that are to be added to an Appearance.
Subclass of:	None
Stereotype:	«DataType»

ADEOfGeoreferencedTexture

Definition:	ADEOfGeoreferencedTexture acts as a hook to define properties within an ADE that are to be added to a GeoreferencedTexture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfParameterizedTexture

Definition:	ADEOfParameterizedTexture acts as a hook to define properties within an ADE that are to be added to a ParameterizedTexture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfX3DMaterial

Definition:	ADEOfX3DMaterial acts as a hook to define properties within an ADE that are to be added to an X3DMaterial.
Subclass of:	None
Stereotype:	«DataType»

TexCoordGen

Definition:	TexCoordGen defines texture parameterization using a transformation matrix.
Subclass of:	None
Stereotype:	«DataType»

Role name	Target class and multiplicity	Definition
crs	SC_CRS [0..1]	Relates to the coordinate reference system of the transformation matrix.

Attribute	Value type and multiplicity	Definition
worldToTexture	TransformationMatrix3x4 [1..1]	Specifies the 3x4 transformation matrix that defines the transformation between world coordinates and texture coordinates.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TexCoordList

Definition:	TexCoordList defines texture parameterization using texture coordinates.
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
textureCoordinates	DoubleList [1..*]	Specifies the coordinates of texture used for parameterization. The texture coordinates are provided separately for each LinearRing of the surface geometry object.
ring	URI [1..*]	Specifies the URIs that point to the LinearRings that are parameterized using the given texture coordinates.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.2.3. Basic Types

Color

Definition:	Color is a list of three double values between 0 and 1 defining an RGB color value.
Subclass of:	DoubleBetween0and1List
Stereotype:	«BasicType»
Constraint:	lengthOfList (OCL): inv: list→size() = 3

ColorPlusOpacity

Definition:	Color is a list of four double values between 0 and 1 defining an RGBA color value. Opacity value of 0 means transparent.
Subclass of:	DoubleBetween0and1List
Stereotype:	«BasicType»
Constraint:	lengthOfList (OCL): inv: list→size() = 3 or list→size() = 4

E.2.4. Unions

none

E.2.5. Code Lists

none

E.2.6. Enumerations

TextureType

Definition:	TextureType enumerates the different texture types.
StereoType:	<<Enumeration>>

Literal value	Definition
specific	Indicates that the texture is specific to a single surface.
typical	Indicates that the texture is characteristic of a surface and can be used repeatedly.
unknown	Indicates that the texture type is not known.

WrapMode

Definition:	WrapMode enumerates the different fill modes for textures.
StereoType:	<<Enumeration>>

Literal value	Definition
none	Indicates that the texture is applied to the surface "as is". The part of the surface that is not covered by the texture is shown fully transparent. [cf. COLLADA]
wrap	Indicates that the texture is repeated until the surface is fully covered. [cf. COLLADA]
mirror	Indicates that the texture is repeated and mirrored. [cf. COLLADA]
clamp	Indicates that the texture is stretched to the edges of the surface. [cf. COLLADA]
border	Indicates that the texture is applied to the surface "as is". The part of the surface that is not covered by the texture is filled with the RGBA color that is specified in the attribute borderColor. [cf. COLLADA]

E.3. CityFurniture

Description:	The CityFurniture module supports representation of city furniture objects. City furniture objects are immovable objects like lanterns, traffic signs, advertising columns, benches, or bus stops that can be found in traffic areas, residential areas, on squares, or in built-up areas.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.3.1. Classes

CityFurniture

Definition:	CityFurniture is an object or piece of equipment installed in the outdoor environment for various purposes. Examples include street signs, traffic signals, street lamps, benches, fountains.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	CityFurnitureClassValue [0..1]	Indicates the specific type of the CityFurniture.
function	CityFurnitureFunctionValue [0..*]	Specifies the intended purposes of the CityFurniture.
usage	CityFurnitureUsageValue [0..*]	Specifies the actual uses of the CityFurniture.
adeOfCityFurniture	ADEOfCityFurniture [0..*]	Augments the CityFurniture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.3.2. Data Types

ADEOfCityFurniture

Definition:	ADEOfCityFurniture acts as a hook to define properties within an ADE that are to be added to a CityFurniture.
Subclass of:	None
Stereotype:	«DataType»

E.3.3. Basic Types

none

E.3.4. Unions

none

E.3.5. Code Lists

CityFurnitureClassValue	
Definition:	CityFurnitureClassValue is a code list used to further classify a CityFurniture.
Stereotype:	«CodeList»

CityFurnitureFunctionValue	
Definition:	CityFurnitureFunctionValue is a code list that enumerates the different purposes of a CityFurniture.
Stereotype:	«CodeList»

CityFurnitureUsageValue	
Definition:	CityFurnitureUsageValue is a code list that enumerates the different uses of a CityFurniture.
Stereotype:	«CodeList»

E.3.6. Enumerations

none

E.4. CityObjectGroup

Description:	The CityObjectGroup module supports grouping of city objects. Arbitrary city objects may be aggregated in groups according to user-defined criteria. A group may be further classified by application-specific attributes.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.4.1. Classes

CityObjectGroup

Definition:	A CityObjectGroup represents an application-specific aggregation of city objects according to some user-defined criteria. Examples for groups are the buildings in a specific region, the result of a query, or objects put together for visualization purposes. Each member of a group may be qualified by a role name, reflecting the role each city object plays in the context of the group.
Subclass of:	AbstractLogicalSpace
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
parent	AbstractCityObject [0..1]	Relates to a city object to which the CityObjectGroup belongs.
groupMember	AbstractCityObject [*]	Relates to the city objects that are part of the CityObjectGroup.

Attribute	Value type and multiplicity	Definition
class	CityObjectGroupClassValue [0..1]	Indicates the specific type of the CityObjectGroup.
function	CityObjectGroupFunctionValue [0..*]	Specifies the intended purposes of the CityObjectGroup.
usage	CityObjectGroupUsageValue [0..*]	Specifies the actual usages of the CityObjectGroup.
adeOfCityObjectGroup	ADEOfCityObjectGroup [0..*]	Augments the CityObjectGroup with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Role

Definition:	Role qualifies the function of a city object within the CityObjectGroup.
Subclass of:	None
Stereotype:	«ObjectType»

Attribute	Value type and multiplicity	Definition
role	CharacterString [0..1]	Describes the role the city object plays within the CityObjectGroup.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.4.2. Data Types

ADEOfCityObjectGroup

Definition:	ADEOfCityObjectGroup acts as a hook to define properties within an ADE that are to be added to a CityObjectGroup.
Subclass of:	None
Stereotype:	«DataType»

E.4.3. Basic Types

none

E.4.4. Unions

none

E.4.5. Code Lists

CityObjectGroupClassValue

Definition:	CityObjectGroupClassValue is a code list used to further classify a CityObjectGroup.
Stereotype:	«CodeList»

CityObjectGroupFunctionValue

Definition:	CityObjectGroupFunctionValue is a code list that enumerates the different purposes of a CityObjectGroup.
Stereotype:	«CodeList»

CityObjectGroupUsageValue

Definition:	CityObjectGroupUsageValue is a code list that enumerates the different uses of a CityObjectGroup.
Stereotype:	«CodeList»

E.4.6. Enumerations

none

E.5. Dynamizer

Description:	The Dynamizer module supports the injection of timeseries data for individual attributes of CityGML features. Timeseries data can either be retrieved from external Sensor APIs (e.g. OGC SensorThings API, OGC Sensor Observation Services, MQTT, proprietary platforms), external standardized timeseries files (e.g. OGC TimeseriesML or OGC Observations & Measurements), external tabulated files (e.g CSV) or can be represented inline as basic time-value pairs.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.5.1. Classes

AbstractAtomicTimeseries

Definition:	AbstractAtomicTimeseries represents the attributes and relationships that are common to all kinds of atomic timeseries (GenericTimeseries, TabulatedFileTimeseries, StandardFileTimeseries). An atomic timeseries represents time-varying data of a specific data type for a single contiguous time interval.
Subclass of:	AbstractTimeseries
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
observationProperty	CharacterString [1..1]	Specifies the phenomenon for which the atomic timeseries provides observation values.
uom	CharacterString [0..1]	Specifies the unit of measurement of the observation values.
adeOfAbstractAtomicTimeseries	ADEOfAbstractAtomicTimeseries [0..*]	Augments AbstractAtomicTimeseries with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractTimeseries

Definition:	AbstractTimeseries is the abstract superclass representing any type of timeseries data.
Subclass of:	AbstractFeature
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
firstTimestamp	TM_Position [0..1]	Specifies the beginning of the timeseries.
lastTimestamp	TM_Position [0..1]	Specifies the end of the timeseries.
adeOfAbstractTimeseries	ADEOfAbstractTimeseries [0..*]	Augments AbstractTimeseries with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

CompositeTimeseries

Definition:	A CompositeTimeseries is a (possibly recursive) aggregation of atomic and composite timeseries. The components of a composite timeseries must have non-overlapping time intervals.
Subclass of:	AbstractTimeseries
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
component	TimeseriesComponent [1..*]	Relates to the atomic and composite timeseries that are part of the CompositeTimeseries. The referenced timeseries are sequentially ordered.

Attribute	Value type and multiplicity	Definition
adeOfCompositeTimeseries	ADEOfCompositeTimeseries [0..*]	Augments the CompositeTimeseries with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Dynamizer

Definition:	A Dynamizer is an object that injects timeseries data for an individual attribute of the city object in which it is included. The timeseries data overrides the static value of the referenced city object attribute in order to represent dynamic (time-dependent) variations of its value.
Subclass of:	AbstractDynamizer
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
dynamicData	AbstractTimeseries [0..1]	Relates to the timeseries data that is given either inline within a CityGML dataset or by a link to an external file containing timeseries data.
sensorConnection	SensorConnection [0..1]	Relates to the sensor API that delivers timeseries data.

Attribute	Value type and multiplicity	Definition
attributeRef	CharacterString [1..1]	Specifies the attribute of a CityGML feature whose value is overridden or replaced by the (dynamic) values specified by the Dynamizer.
startTime	TM_Position [0..1]	Specifies the beginning of the time span for which the Dynamizer provides dynamic values.
endTime	TM_Position [0..1]	Specifies the end of the time span for which the Dynamizer provides dynamic values.
adeOfDynamizer	ADEOfDynamizer [0..*]	Augments the Dynamizer with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GenericTimeseries

Definition:	A GenericTimeseries represents time-varying data in the form of embedded time-value-pairs of a specific data type for a single contiguous time interval.
Subclass of:	AbstractAtomicTimeseries
Stereotype:	«FeatureType»
Constraint:	dataTypeOfValue (OCL): inv: if valueType = TimeseriesTypeValue::integer then TimeValuePair→forAll(c | c.intValue→size()=1) else if valueType = TimeseriesTypeValue::double then TimeValuePair→forAll(c | c.doubleValue→size()=1) else if valueType = TimeseriesTypeValue::string then TimeValuePair→forAll(c | c.stringValue→size()=1) else if valueType = TimeseriesTypeValue::geometry then TimeValuePair→forAll(c | c.geometryValue→size()=1) else if valueType = TimeseriesTypeValue::uri then TimeValuePair→forAll(c | c.uriValue→size()=1) else if valueType = TimeseriesTypeValue::bool then TimeValuePair→forAll(c | c.boolValue→size()=1) else if valueType = TimeseriesTypeValue::implicitGeometry then TimeValuePair→forAll(c | c.implicitGeometryValue→size()=1) else TimeValuePair→forAll(c | c.appearanceValue→size()=1)

Role name	Target class and multiplicity	Definition
timeValuePair	TimeValuePair [1..*]	Relates to the time-value-pairs that are part of the GenericTimeseries.

Attribute	Value type and multiplicity	Definition
valueType	TimeseriesTypeValue [1..1]	Indicates the specific type of all time-value-pairs that are part of the GenericTimeseries.
adeOfGenericTimeseries	ADEOfGenericTimeseries [0..*]	Augments the GenericTimeseries with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

StandardFileTimeseries

Definition:	A StandardFileTimeseries represents time-varying data for a single contiguous time interval. The data is provided in an external file referenced in the StandardFileTimeseries. The data within the external file is encoded according to a dedicated format for the representation of timeseries data such as using the OGC TimeseriesML or OGC Observations & Measurements Standard. The data type of the data has to be specified within the external file.
Subclass of:	AbstractAtomicTimeseries
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
fileLocation	URI [1..1]	Specifies the URI that points to the external timeseries file.
fileType	StandardFileTypeValue [1..1]	Specifies the format used to represent the timeseries data.
mimeType	MimeTypeValue [0..1]	Specifies the MIME type of the external timeseries file.
adeOfStandardFileTimeseries	ADEOfStandardFileTimeseries [0..*]	Augments the StandardFileTimeseries with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TabulatedFileTimeseries

Definition:	A TabulatedFileTimeseries represents time-varying data of a specific data type for a single contiguous time interval. The data is provided in an external file referenced in the TabulatedFileTimeseries. The file contains table structured data using an appropriate file format such as comma-separated values (CSV), Microsoft Excel (XLSX) or Google Spreadsheet. The timestamps and the values are given in specific columns of the table. Each row represents a single time-value-pair. A subset of rows can be selected using the idColumn and idValue attributes.
Subclass of:	AbstractAtomicTimeseries
Stereotype:	«FeatureType»
Constraint:	columnNumberOrColumnName (OCL): inv: (timeColumnNo→notEmpty() or timeColumnName→notEmpty()) and (valueColumnNo→notEmpty() or valueColumnName→notEmpty()) and (idValue→notEmpty() implies idColumnNo→notEmpty() or + idColumnName→notEmpty())

Attribute	Value type and multiplicity	Definition
fileLocation	URI [1..1]	Specifies the URI that points to the external timeseries file.
fileType	TabulatedFileTypeValue [1..1]	Specifies the format used to represent the timeseries data.
mimeType	MimeTypeValue [0..1]	Specifies the MIME type of the external timeseries file.
valueType	TimeseriesTypeValue [1..1]	Indicates the specific type of the timeseries data.
numberOfHeaderLines	Integer [0..1]	Indicates the number of lines at the beginning of the tabulated file that represent headers.
fieldSeparator	CharacterString [1..1]	Indicates which symbol is used to separate the individual values in the tabulated file.
decimalSymbol	Character [0..1]	Indicates which symbol is used to separate the integer part from the fractional part of a decimal number.
idColumnNo	Integer [0..1]	Specifies the number of the column that stores the identifier of the time-value-pair.
idColumnName	CharacterString [0..1]	Specifies the name of the column that stores the identifier of the time-value-pair.
idValue	CharacterString [0..1]	Specifies the value of the identifier for which the time-value-pairs are to be selected.
timeColumnNo	Integer [0..1]	Specifies the number of the column that stores the timestamp of the time-value-pair.
timeColumnName	CharacterString [0..1]	Specifies the name of the column that stores the timestamp of the time-value-pair.
valueColumnNo	Integer [0..1]	Specifies the number of the column that stores the value of the time-value-pair.
valueColumnName	CharacterString [0..1]	Specifies the name of the column that stores the value of the time-value-pair.
adeOfTabulatedFileTimeseries	ADEOfTabulatedFileTimeseries [0..*]	Augments the TabulatedFileTimeseries with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.5.2. Data Types

ADEOfAbstractAtomicTimeseries

Definition:	ADEOfAbstractAtomicTimeseries acts as a hook to define properties within an ADE that are to be added to AbstractAtomicTimeseries.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractTimeseries

Definition:	ADEOfAbstractTimeseries acts as a hook to define properties within an ADE that are to be added to AbstractTimeseries.
Subclass of:	None
Stereotype:	«DataType»

ADEOfCompositeTimeseries

Definition:	ADEOfCompositeTimeseries acts as a hook to define properties within an ADE that are to be added to a CompositeTimeseries.
Subclass of:	None
Stereotype:	«DataType»

ADEOfDynamizer

Definition:	ADEOfDynamizer acts as a hook to define properties within an ADE that are to be added to a Dynamizer.
Subclass of:	None
Stereotype:	«DataType»

ADEOfGenericTimeseries

Definition:	ADEOfGenericTimeseries acts as a hook to define properties within an ADE that are to be added to a GenericTimeseries.
Subclass of:	None
Stereotype:	«DataType»

ADEOfStandardFileTimeseries

Definition:	ADEOfStandardFileTimeseries acts as a hook to define properties within an ADE that are to be added to a StandardFileTimeseries.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTabulatedFileTimeseries

Definition:	ADEOfTabulatedFileTimeseries acts as a hook to define properties within an ADE that are to be added to a TabulatedFileTimeseries.
Subclass of:	None
Stereotype:	«DataType»

SensorConnection

Definition:	A SensorConnection provides all details that are required to retrieve a specific datastream from an external sensor web service. This data type comprises the service type (e.g. OGC SensorThings API, OGC Sensor Observation Services, MQTT, proprietary platforms), the URL of the sensor service, the identifier for the sensor or thing, and its observed property as well as information about the required authentication method.
Subclass of:	None
Stereotype:	«DataType»

Role name	Target class and multiplicity	Definition
sensorLocation	AbstractCityObject [0..1]	Relates the sensor to the city object where it is located.

Attribute	Value type and multiplicity	Definition
connectionType	SensorConnectionTypeValue [1..1]	Indicates the type of Sensor API to which the SensorConnection refers.
observationProperty	CharacterString [1..1]	Specifies the phenomenon for which the SensorConnection provides observations.
uom	CharacterString [0..1]	Specifies the unit of measurement of the observations.
sensorID	CharacterString [0..1]	Specifies the unique identifier of the sensor from which the SensorConnection retrieves observations.
sensorName	CharacterString [0..1]	Specifies the name of the sensor from which the SensorConnection retrieves observations.
observationID	CharacterString [0..1]	Specifies the unique identifier of the observation that is retrieved by the SensorConnection.
datastreamID	CharacterString [0..1]	Specifies the datastream that is retrieved by the SensorConnection.
baseURL	URI [0..1]	Specifies the base URL of the Sensor API request.
authType	AuthenticationTypeValue [0..1]	Specifies the type of authentication required to be able to access the Sensor API.
mqttServer	CharacterString [0..1]	Specifies the name of the MQTT Server. This attribute is relevant when the MQTT Protocol is used to connect to a Sensor API.
mqttTopic	CharacterString [0..1]	Names the specific datastream that is retrieved by the SensorConnection. This attribute is relevant when the MQTT Protocol is used to connect to a Sensor API.
linkToObservation	CharacterString [0..1]	Specifies the complete URL to the observation request.
linkToSensorDescription	CharacterString [0..1]	Specifies the complete URL to the sensor description request.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TimeseriesComponent

Definition:	TimeseriesComponent represents an element of a CompositeTimeseries.
Subclass of:	None
Stereotype:	«DataType»

Role name	Target class and multiplicity	Definition
timeseries	AbstractTimeseries [1..1]	Relates a timeseries to the TimeseriesComponent.

Attribute	Value type and multiplicity	Definition
repetitions	Integer [1..1]	Specifies how often the timeseries that is referenced by the TimeseriesComponent should be iterated.
additionalGap	TM_Duration [0..1]	Specifies how much extra time is added after all repetitions as an additional gap.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TimeValuePair

Definition:	A TimeValuePair represents a value that is valid for a given timepoint. For each TimeValuePair, only one of the value properties can be used mutually exclusive. Which value property has to be provided depends on the selected value type in the GenericTimeSeries feature, in which the TimeValuePair is included.
Subclass of:	None
Stereotype:	«DataType»
Constraint:	singleValue (OCL): inv: intValue→size() + doubleValue→size() + stringValue→size() + geometryValue→size() + uriValue→size() + boolValue→size() + implicitGeometryValue→size() + appearanceValue→size() = 1

Attribute	Value type and multiplicity	Definition
timestamp	TM_Position [1..1]	Specifies the timepoint at which the value of the TimeValuePair is valid.
intValue	Integer [0..1]	Specifies the "Integer" value of the TimeValuePair.
doubleValue	Real [0..1]	Specifies the "Double" value of the TimeValuePair.
stringValue	CharacterString [0..1]	Specifies the "String" value of the TimeValuePair.
geometryValue	GM_Object [0..1]	Specifies the geometry value of the TimeValuePair.
uriValue	URI [0..1]	Specifies the "URI" value of the TimeValuePair.
boolValue	Boolean [0..1]	Specifies the "Boolean" value of the TimeValuePair.
implicitGeometryValue	ImplicitGeometry [0..1]	Specifies the "ImplicitGeometry" value of the TimeValuePair.
appearanceValue	AbstractAppearance [0..1]	Specifies the "Appearance" value of the TimeValuePair.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.5.3. Basic Types

none

E.5.4. Unions

none

E.5.5. Code Lists

AuthenticationTypeValue

Definition:	AuthenticationTypeValue is a code list used to specify the authentication method to be used to access the referenced sensor service. Each value provides enough information such that a software application could determine the required access credentials.
Stereotype:	«CodeList»

SensorConnectionTypeValue

Definition:	SensorConnectionTypeValue is a code list used to specify the type of the referenced sensor service. Each value provides enough information such that a software application would be able to identify the API type and version.
Stereotype:	«CodeList»

StandardFileTypeValue

Definition:	StandardFileTypeValue is a code list used to specify the type of the referenced external timeseries data file. Each value provides information about the standard and version.
Stereotype:	«CodeList»

TabulatedFileTypeValue

Definition:	TabulatedFileTypeValue is a code list used to specify the data format of the referenced external tabulated data file.
Stereotype:	«CodeList»

E.5.6. Enumerations

TimeseriesTypeValue

Definition:	TimeseriesTypeValue enumerates the possible value types for GenericTimeseries and TimeValuePair.
StereoType:	<<Enumeration>>

Literal value	Definition
int	Indicates that the values of the GenericTimeseries are of type "Integer".
double	Indicates that the values of the GenericTimeseries are of type "Double".
string	Indicates that the values of the GenericTimeseries are of type "String".
geometry	Indicates that the values of the GenericTimeseries are geometries.
uri	Indicates that the values of the GenericTimeseries are of type "URI".
bool	Indicates that the values of the GenericTimeseries are of type "Boolean".
implicitGeometry	Indicates that the values of the GenericTimeseries are of type "ImplicitGeometry".
appearance	Indicates that the values of the GenericTimeseries are of type "Appearance".

E.6. Generics

Description:	The Generics module supports application-specific extensions to the CityGML data model. These extensions may be used to model and exchange additional attributes and features not covered by the predefined thematic classes of CityGML. Generic extensions shall only be used if appropriate thematic classes or attributes are not provided by any other CityGML module.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.6.1. Classes

GenericLogicalSpace

Definition:	A GenericLogicalSpace is a space that is not represented by any explicitly modelled AbstractLogicalSpace subclass within CityGML.
Subclass of:	AbstractLogicalSpace
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	GenericLogicalSpaceClassValue [0..1]	Indicates the specific type of the GenericLogicalSpace.
function	GenericLogicalSpaceFunctionValue [0..*]	Specifies the intended purposes of the GenericLogicalSpace.
usage	GenericLogicalSpaceUsageValue [0..*]	Specifies the actual uses of the GenericLogicalSpace.
adeOfGenericLogicalSpace	ADEOfGenericLogicalSpace [0..*]	Augments the GenericLogicalSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GenericOccupiedSpace

Definition:	A GenericOccupiedSpace is a space that is not represented by any explicitly modelled AbstractOccupiedSpace subclass within CityGML.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	GenericOccupiedSpaceClassValue [0..1]	Indicates the specific type of the GenericOccupiedSpace.
function	GenericOccupiedSpaceFunctionValue [0..*]	Specifies the intended purposes of the GenericOccupiedSpace.
usage	GenericOccupiedSpaceUsageValue [0..*]	Specifies the actual uses of the GenericOccupiedSpace.
adeOfGenericOccupiedSpace	ADEOfGenericOccupiedSpace [0..*]	Augments the GenericOccupiedSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GenericThematicSurface

Definition:	A GenericThematicSurface is a surface that is not represented by any explicitly modelled AbstractThematicSurface subclass within CityGML.
Subclass of:	AbstractThematicSurface
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	GenericThematicSurfaceClassValue [0..1]	Indicates the specific type of the GenericThematicSurface.
function	GenericThematicSurfaceFunctionValue [0..*]	Specifies the intended purposes of the GenericThematicSurface.
usage	GenericThematicSurfaceUsageValue [0..*]	Specifies the actual uses of the GenericThematicSurface.
adeOfGenericThematicSurface	ADEOfGenericThematicSurface [0..*]	Augments the GenericThematicSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GenericUnoccupiedSpace

Definition:	A GenericUnoccupiedSpace is a space that is not represented by any explicitly modelled AbstractUnoccupiedSpace subclass within CityGML.
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	GenericUnoccupiedSpaceClassValue [0..1]	Indicates the specific type of the GenericUnoccupiedSpace.
function	GenericUnoccupiedSpaceFunctionValue [0..*]	Specifies the intended purposes of the GenericUnoccupiedSpace.
usage	GenericUnoccupiedSpaceUsageValue [0..*]	Specifies the actual uses of the GenericUnoccupiedSpace.
adeOfGenericUnoccupiedSpace	ADEOfGenericUnoccupiedSpace [0..*]	Augments the GenericUnoccupiedSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.6.2. Data Types

ADEOfGenericLogicalSpace

Definition:	ADEOfGenericLogicalSpace acts as a hook to define properties within an ADE that are to be added to a GenericLogicalSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfGenericOccupiedSpace

Definition:	ADEOfGenericOccupiedSpace acts as a hook to define properties within an ADE that are to be added to a GenericOccupiedSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfGenericThematicSurface

Definition:	ADEOfGenericThematicSurface acts as a hook to define properties within an ADE that are to be added to a GenericThematicSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfGenericUnoccupiedSpace

Definition:	ADEOfGenericUnoccupiedSpace acts as a hook to define properties within an ADE that are to be added to a GenericUnoccupiedSpace.
Subclass of:	None
Stereotype:	«DataType»

CodeAttribute

Definition:	CodeAttribute is a data type used to define generic attributes of type "Code".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the CodeAttribute.
value	Code [1..1]	Specifies the "Code" value.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

DateAttribute

Definition:	DateAttribute is a data type used to define generic attributes of type "Date".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the DateAttribute.
value	Date [1..1]	Specifies the "Date" value.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

DoubleAttribute

Definition:	DoubleAttribute is a data type used to define generic attributes of type "Double".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the DoubleAttribute.
value	Real [1..1]	Specifies the "Double" value.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GenericAttributeSet

Definition:	A GenericAttributeSet is a named collection of generic attributes.
Subclass of:	None
Stereotype:	«DataType»

Role name	Target class and multiplicity	Definition
genericAttribute	AbstractGenericAttribute [1..*]	Relates to the generic attributes that are part of the GenericAttributeSet.

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the GenericAttributeSet.
codeSpace	URI [0..1]	Associates the GenericAttributeSet with an authority that maintains the collection of generic attributes.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

IntAttribute

Definition:	IntAttribute is a data type used to define generic attributes of type "Integer".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the IntAttribute.
value	Integer [1..1]	Specifies the "Integer" value.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

MeasureAttribute

Definition:	MeasureAttribute is a data type used to define generic attributes of type "Measure".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the MeasureAttribute.
value	Measure [1..1]	Specifies the value of the MeasureAttribute. The value is of type "Measure", which can additionally provide the units of measure. [cf. ISO 19103]

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

StringAttribute

Definition:	StringAttribute is a data type used to define generic attributes of type "String".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the StringAttribute.
value	CharacterString [1..1]	Specifies the "String" value.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

UriAttribute

Definition:	UriAttribute is a data type used to define generic attributes of type "URI".
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
name	CharacterString [1..1]	Specifies the name of the UriAttribute.
value	URI [1..1]	Specifies the "URI" value.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.6.3. Basic Types

none

E.6.4. Unions

none

E.6.5. Code Lists

GenericLogicalSpaceClassValue

Definition:	GenericLogicalSpaceClassValue is a code list used to further classify a GenericLogicalSpace.
Stereotype:	«CodeList»

GenericLogicalSpaceFunctionValue

Definition:	GenericLogicalSpaceFunctionValue is a code list that enumerates the different purposes of a GenericLogicalSpace.
Stereotype:	«CodeList»

GenericLogicalSpaceUsageValue

Definition:	GenericLogicalSpaceUsageValue is a code list that enumerates the different uses of a GenericLogicalSpace.
Stereotype:	«CodeList»

GenericOccupiedSpaceClassValue

Definition:	GenericOccupiedSpaceClassValue is a code list used to further classify a GenericOccupiedSpace.
Stereotype:	«CodeList»

GenericOccupiedSpaceFunctionValue

Definition:	GenericOccupiedSpaceFunctionValue is a code list that enumerates the different purposes of a GenericOccupiedSpace.
Stereotype:	«CodeList»

GenericOccupiedSpaceUsageValue

Definition:	GenericOccupiedSpaceUsageValue is a code list that enumerates the different uses of a GenericOccupiedSpace.
Stereotype:	«CodeList»

GenericThematicSurfaceClassValue

Definition:	GenericThematicSurfaceClassValue is a code list used to further classify a GenericThematicSurface.
Stereotype:	«CodeList»

GenericThematicSurfaceFunctionValue

Definition:	GenericThematicSurfaceFunctionValue is a code list that enumerates the different purposes of a GenericThematicSurface.
Stereotype:	«CodeList»

GenericThematicSurfaceUsageValue

Definition:	GenericThematicSurfaceUsageValue is a code list that enumerates the different uses of a GenericThematicSurface.
Stereotype:	«CodeList»

GenericUnoccupiedSpaceClassValue

Definition:	GenericUnoccupiedSpaceClassValue is a code list used to further classify a GenericUnoccupiedSpace.
Stereotype:	«CodeList»

GenericUnoccupiedSpaceFunctionValue

Definition:	GenericUnoccupiedSpaceFunctionValue is a code list that enumerates the different purposes of a GenericUnoccupiedSpace.
Stereotype:	«CodeList»

GenericUnoccupiedSpaceUsageValue

Definition:	GenericUnoccupiedSpaceUsageValue is a code list that enumerates the different uses of a GenericUnoccupiedSpace.
Stereotype:	«CodeList»

E.6.6. Enumerations

none

E.7. LandUse

Description:	The LandUse module supports representation of areas of the earth's surface dedicated to a specific land use.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.7.1. Classes

LandUse

Definition:	A LandUse object is an area of the earth's surface dedicated to a specific land use or having a specific land cover with or without vegetation, such as sand, rock, mud flats, forest, grasslands, or wetlands.
Subclass of:	AbstractThematicSurface
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	LandUseClassValue [0..1]	Indicates the specific type of the LandUse.
function	LandUseFunctionValue [0..*]	Specifies the intended purposes of the LandUse.
usage	LandUseUsageValue [0..*]	Specifies the actual uses of the LandUse.
adeOfLandUse	ADEOfLandUse [0..*]	Augments the LandUse with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.7.2. Data Types

ADEOfLandUse	
Definition:	ADEOfLandUse acts as a hook to define properties within an ADE that are to be added to a LandUse.
Subclass of:	None
Stereotype:	«DataType»

E.7.3. Basic Types

none

E.7.4. Unions

none

E.7.5. Code Lists

LandUseClassValue	
Definition:	LandUseClassValue is a code list used to further classify a LandUse.
Stereotype:	«CodeList»

LandUseFunctionValue	
Definition:	LandUseFunctionValue is a code list that enumerates the different purposes of a LandUse.
Stereotype:	«CodeList»

LandUseUsageValue	
Definition:	LandUseUsageValue is a code list that enumerates the different uses of a LandUse.
Stereotype:	«CodeList»

E.7.6. Enumerations

none

E.8. PointCloud

Description:	The PointCloud module supports representation of CityGML features by a collection of points.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.8.1. Classes

PointCloud

Definition:	A PointCloud is an unordered collection of points that is a sampling of the geometry of a space or space boundary.
Subclass of:	AbstractPointCloud
Stereotype:	«FeatureType»
Constraint:	inlineOrExternalPointCloud (OCL): inv: (points→notEmpty() and mimeType→isEmpty() and pointFile→isEmpty() and pointFileSrsName→isEmpty()) xor (points→isEmpty() and mimeType→notEmpty() and pointFile→notEmpty())

Role name	Target class and multiplicity	Definition
points	GM_MultiPoint [0..1]	Relates to the 3D MultiPoint geometry of the PointCloud stored inline with the city model.

Attribute	Value type and multiplicity	Definition
mimeType	MimeTypeValue [0..1]	Specifies the MIME type of the external point cloud file.
pointFile	URI [0..1]	Specifies the URI that points to the external point cloud file.
pointFileSrsName	CharacterString [0..1]	Indicates the coordinate reference system used by the external point cloud file.
adeOfPointCloud	ADEOfPointCloud [0..*]	Augments the PointCloud with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.8.2. Data Types

ADEOfPointCloud

Definition:	ADEOfPointCloud acts as a hook to define properties within an ADE that are to be added to a PointCloud.
Subclass of:	None
Stereotype:	«DataType»

E.8.3. Basic Types

none

E.8.4. Unions

none

E.8.5. Code Lists

none

E.8.6. Enumerations

none

E.9. Relief

Description:	The Relief module supports representation of the terrain. CityGML supports terrain representations at different levels of detail, reflecting different accuracies or resolutions. Terrain may be specified as a regular raster or grid, as a TIN, by break lines, and/or by mass points.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.9.1. Classes

AbstractReliefComponent

Definition:	An AbstractReliefComponent represents an element of the terrain surface - either a TIN, a raster or grid, mass points or break lines.
Subclass of:	AbstractSpaceBoundary
Stereotype:	«FeatureType»
Constraint:	polygonGeometry (OCL): inv: extent.patch→size()=1 and extent.patch→forAll(oclIsKindOf(GM_Polygon))

Role name	Target class and multiplicity	Definition
extent	GM_Surface [0..1]	Indicates the geometrical extent of the terrain component. The geometrical extent is provided as a 2D Surface geometry.

Attribute	Value type and multiplicity	Definition
lod	IntegerBetween0and3 [1..1]	Indicates the Level of Detail of the terrain component.
adeOfAbstractReliefComponent	ADEOfAbstractReliefComponent [0..*]	Augments AbstractReliefComponent with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BreaklineRelief

Definition:	A BreaklineRelief represents a terrain component with 3D lines. These lines denote break lines or ridge/valley lines.
Subclass of:	AbstractReliefComponent
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
breaklines	GM_MultiCurve [0..1]	Relates to the 3D MultiCurve geometry of the MassPointRelief. This association role is used to represent break lines.
ridgeOrValleyLines	GM_MultiCurve [0..1]	Relates to the 3D MultiCurve geometry of the MassPointRelief. This association role is used to represent ridge or valley lines.

Attribute	Value type and multiplicity	Definition
adeOfBreaklineRelief	ADEOfBreaklineRelief [0..*]	Augments the BreaklineRelief with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

MassPointRelief

Definition:	A MassPointRelief represents a terrain component as a collection of 3D points.
Subclass of:	AbstractReliefComponent
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
pointCloud	AbstractPointCloud [0..1]	Relates to the 3D PointCloud of the MassPointRelief.
reliefPoints	GM_MultiPoint [0..1]	Relates to the 3D MultiPoint geometry of the MassPointRelief.

Attribute	Value type and multiplicity	Definition
adeOfMassPointRelief	ADEOfMassPointRelief [0..*]	Augments the MassPointRelief with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

RasterRelief

Definition:	A RasterRelief represents a terrain component as a regular raster or grid.
Subclass of:	AbstractReliefComponent
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
grid	CV_DiscreteGridPointCoverage [1]	Relates to the DiscreteGridPointCoverage of the RasterRelief.

Attribute	Value type and multiplicity	Definition
adeOfRasterRelief	ADEOfRasterRelief [0..*]	Augments the RasterRelief with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

ReliefFeature

Definition:	A ReliefFeature is a collection of terrain components representing the Earth's surface, also known as the Digital Terrain Model.
Subclass of:	AbstractSpaceBoundary
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
reliefComponent	AbstractReliefComponent [1..*]	Relates to the terrain components that are part of the ReliefFeature.

Attribute	Value type and multiplicity	Definition
lod	IntegerBetween0and3 [1..1]	Indicates the Level of Detail of the ReliefFeature.
adeOfReliefFeature	ADEOfReliefFeature [0..*]	Augments the ReliefFeature with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TINRelief

Definition:	A TINRelief represents a terrain component as a triangulated irregular network.
Subclass of:	AbstractReliefComponent
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
tin	GM_TriangulatedSurface [1]	Relates to the triangulated surface of the TINRelief.

Attribute	Value type and multiplicity	Definition
adeOfTINRelief	ADEOfTINRelief [0..*]	Augments the TINRelief with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.9.2. Data Types

ADEOfAbstractReliefComponent

Definition:	ADEOfAbstractReliefComponent acts as a hook to define properties within an ADE that are to be added to AbstractReliefComponent.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBreaklineRelief

Definition:	ADEOfBreaklineRelief acts as a hook to define properties within an ADE that are to be added to a BreaklineRelief.
Subclass of:	None
Stereotype:	«DataType»

ADEOfMassPointRelief

Definition:	ADEOfMassPointRelief acts as a hook to define properties within an ADE that are to be added to a MassPointRelief.
Subclass of:	None
Stereotype:	«DataType»

ADEOfRasterRelief

Definition:	ADEOfRasterRelief acts as a hook to define properties within an ADE that are to be added to a RasterRelief.
Subclass of:	None
Stereotype:	«DataType»

ADEOfReliefFeature

Definition:	ADEOfReliefFeature acts as a hook to define properties within an ADE that are to be added to a ReliefFeature.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTINRelief

Definition:	ADEOfTINRelief acts as a hook to define properties within an ADE that are to be added to a TINRelief.
Subclass of:	None
Stereotype:	«DataType»

E.9.3. Basic Types

none

E.9.4. Unions

none

E.9.5. Code Lists

none

E.9.6. Enumerations

none

E.10. Transportation

Description:	The Transportation module supports representation of the transportation infrastructure. Transportation features include roads, tracks, waterways, railways, and squares. Transportation features may be represented as a network and/or as a collection of spaces or surface elements embedded in a three-dimensional space.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.10.1. Classes

AbstractTransportationSpace

Definition:	AbstractTransportationSpace is the abstract superclass of transportation objects such as Roads, Tracks, Railways, Waterways or Squares.
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
marking	Marking [*]	Relates to the markings that are part of the transportation space.
trafficSpace	TrafficSpace [*]	Relates to the traffic spaces that are part of the transportation space.
auxiliaryTrafficSpace	AuxiliaryTrafficSpace [*]	Relates to the auxiliary traffic spaces that are part of the transportation space.
hole	Hole [*]	Relates to the holes that are part of the transportation space.

Attribute	Value type and multiplicity	Definition
trafficDirection	TrafficDirectionValue [0..1]	Indicates the direction of traffic flow relative to the corresponding linear geometry representation.
occupancy	Occupancy [0..*]	Provides information on the residency of persons, vehicles, or other moving features in the transportation space.
adeOfAbstractTransportationSpace	ADEOfAbstractTransportationSpace [0..*]	Augments AbstractTransportationSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AuxiliaryTrafficArea

Definition:	An AuxiliaryTrafficArea is the ground surface of an AuxiliaryTrafficSpace.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	AuxiliaryTrafficAreaClassValue [0..1]	Indicates the specific type of the AuxiliaryTrafficArea.
function	AuxiliaryTrafficAreaFunctionValue [0..*]	Specifies the intended purposes of the AuxiliaryTrafficArea.
usage	AuxiliaryTrafficAreaUsageValue [0..*]	Specifies the actual uses of the AuxiliaryTrafficArea.
surfaceMaterial	SurfaceMaterialValue [0..1]	Specifies the type of pavement of the AuxiliaryTrafficArea.
adeOfAuxiliaryTrafficArea	ADEOfAuxiliaryTrafficArea [0..*]	Augments the AuxiliaryTrafficArea with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AuxiliaryTrafficSpace

Definition:	An AuxiliaryTrafficSpace is a space within the transportation space not intended for traffic purposes.
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	AuxiliaryTrafficArea [*]	Relates to the auxiliary traffic areas that bound the AuxiliaryTrafficSpace. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	AuxiliaryTrafficSpaceClassValue [0..1]	Indicates the specific type of the AuxiliaryTrafficSpace.
function	AuxiliaryTrafficSpaceFunctionValue [0..*]	Specifies the intended purposes of the AuxiliaryTrafficSpace.
usage	AuxiliaryTrafficSpaceUsageValue [0..*]	Specifies the actual uses of the AuxiliaryTrafficSpace.
granularity	GranularityValue [1..1]	Defines whether auxiliary traffic spaces are represented by individual ways or by individual lanes, depending on the desired level of spatial and semantic decomposition.
adeOfAuxiliaryTrafficSpace	ADEOfAuxiliaryTrafficSpace [0..*]	Augments the AuxiliaryTrafficSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

ClearanceSpace

Definition:	A ClearanceSpace represents the actual free space above a TrafficArea within which a mobile object can move without contacting an obstruction.
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	ClearanceSpaceClassValue [0..*]	Indicates the specific type of the ClearanceSpace.
adeOfClearanceSpace	ADEOfClearanceSpace [0..*]	Augments the ClearanceSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Hole

Definition:	A Hole is an opening in the surface of a Road, Track or Square such as road damages, manholes or drains. Holes can span multiple transportation objects.
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the Hole. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	HoleClassValue [0..1]	Indicates the specific type of the Hole.
adeOfHole	ADEOfHole [0..*]	Augments the Hole with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

HoleSurface

Definition:	A HoleSurface is a representation of the ground surface of a hole.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfHoleSurface	ADEOfHoleSurface [0..*]	Augments the HoleSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Intersection

Definition:	An Intersection is a transportation space that is a shared segment of multiple Road, Track, Railway, or Waterway objects (e.g. a crossing of two roads or a level crossing of a road and a railway).
Subclass of:	AbstractTransportationSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	IntersectionClassValue [0..1]	Indicates the specific type of the Intersection.
adeOfIntersection	ADEOfIntersection [0..*]	Augments the Intersection with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Marking

Definition:	A Marking is a visible pattern on a transportation area relevant to the structuring or restriction of traffic. Examples are road markings and markings related to railway or waterway traffic.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	MarkingClassValue [0..1]	Indicates the specific type of the Marking.
adeOfMarking	ADEOfMarking [0..*]	Augments the Marking with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Railway

Definition:	A Railway is a transportation space used by wheeled vehicles on rails.
Subclass of:	AbstractTransportationSpace
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
intersection	Intersection [*]	Relates to the intersections that are part of the Railway.
section	Section [*]	Relates to the sections that are part of the Railway.

Attribute	Value type and multiplicity	Definition
class	RailwayClassValue [0..1]	Indicates the specific type of the Railway.
function	RailwayFunctionValue [0..*]	Specifies the intended purposes of the Railway.
usage	RailwayUsageValue [0..*]	Specifies the actual uses of the Railway.
adeOfRailway	ADEOfRailway [0..*]	Augments the Railway with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Road

Definition:	A Road is a transportation space used by vehicles, bicycles and/or pedestrians.
Subclass of:	AbstractTransportationSpace
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
intersection	Intersection [*]	Relates to the intersections that are part of the Road.
section	Section [*]	Relates to the sections that are part of the Road.

Attribute	Value type and multiplicity	Definition
class	RoadClassValue [0..1]	Indicates the specific type of the Road.
function	RoadFunctionValue [0..*]	Specifies the intended purposes of the Road.
usage	RoadUsageValue [0..*]	Specifies the actual uses of the Road.
adeOfRoad	ADEOfRoad [0..*]	Augments the Road with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Section

Definition:	A Section is a transportation space that is a segment of a Road, Railway, Track, or Waterway.
Subclass of:	AbstractTransportationSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	SectionClassValue [0..1]	Indicates the specific type of the Section.
adeOfSection	ADEOfSection [0..*]	Augments the Section with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Square

Definition:	A Square is a transportation space for unrestricted movement for vehicles, bicycles and/or pedestrians. This includes plazas as well as large sealed surfaces such as parking lots.
Subclass of:	AbstractTransportationSpace
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	SquareClassValue [0..1]	Indicates the specific type of the Square.
function	SquareFunctionValue [0..*]	Specifies the intended purposes of the Square.
usage	SquareUsageValue [0..*]	Specifies the actual uses of the Square.
adeOfSquare	ADEOfSquare [0..*]	Augments the Square with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Track

Definition:	A Track is a small path mainly used by pedestrians. Tracks can be segmented into Sections and Intersections.
Subclass of:	AbstractTransportationSpace
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
section	Section [*]	Relates to the sections that are part of the Track.
intersection	Intersection [*]	Relates to the intersections that are part of the Track.

Attribute	Value type and multiplicity	Definition
class	TrackClassValue [0..1]	Indicates the specific type of the Track.
function	TrackFunctionValue [0..*]	Specifies the intended purposes of the Track.
usage	TrackUsageValue [0..*]	Specifies the actual uses of the Track.
adeOfTrack	ADEOfTrack [0..*]	Augments the Track with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TrafficArea

Definition:	A TrafficArea is the ground surface of a TrafficSpace. Traffic areas are the surfaces upon which traffic actually takes place.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	TrafficAreaClassValue [0..1]	Indicates the specific type of the TrafficArea.
function	TrafficAreaFunctionValue [0..*]	Specifies the intended purposes of the TrafficArea.
usage	TrafficAreaUsageValue [0..*]	Specifies the actual uses of the TrafficArea.
surfaceMaterial	SurfaceMaterialValue [0..1]	Specifies the type of pavement of the TrafficArea.
adeOfTrafficArea	ADEOfTrafficArea [0..*]	Augments the TrafficArea with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TrafficSpace

Definition:	A TrafficSpace is a space in which traffic takes place. Traffic includes the movement of entities such as trains, vehicles, pedestrians, ships, or other transportation types.
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
successor	TrafficSpace [*]	Indicates the successor(s) of the TrafficSpace.
clearanceSpace	ClearanceSpace [*]	Relates to the clearance spaces that are part of the TrafficSpace.
predecessor	TrafficSpace [*]	Indicates the predecessor(s) of the TrafficSpace.
boundary	TrafficArea [*]	Relates to the traffic areas that bound the TrafficSpace. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	TrafficSpaceClassValue [0..1]	Indicates the specific type of the TrafficSpace.
function	TrafficSpaceFunctionValue [0..*]	Specifies the intended purposes of the TrafficSpace.
usage	TrafficSpaceUsageValue [0..*]	Specifies the actual uses of the TrafficSpace.
granularity	GranularityValue [1..1]	Defines whether traffic spaces are represented by individual ways or by individual lanes, depending on the desired level of spatial and semantic decomposition.
trafficDirection	TrafficDirectionValue [0..1]	Indicates the direction of traffic flow relative to the corresponding linear geometry representation.
occupancy	Occupancy [0..*]	Provides information on the residency of persons, vehicles, or other moving features in the TrafficSpace.
adeOfTrafficSpace	ADEOfTrafficSpace [0..*]	Augments the TrafficSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Waterway

Definition:	A Waterway is a transportation space used for the movement of vessels upon or within a water body.
Subclass of:	AbstractTransportationSpace
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
intersection	Intersection [*]	Relates to the intersections that are part of the Waterway.
section	Section [*]	Relates to the sections that are part of the Waterway.

Attribute	Value type and multiplicity	Definition
class	WaterwayClassValue [0..1]	Indicates the specific type of the Waterway.
function	WaterwayFunctionValue [0..*]	Specifies the intended purposes of the Waterway.
usage	WaterwayUsageValue [0..*]	Specifies the actual uses of the Waterway.
adeOfWaterway	ADEOfWaterway [0..*]	Augments the Waterway with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.10.2. Data Types

ADEOfAbstractTransportationSpace

Definition:	ADEOfAbstractTransportationSpace acts as a hook to define properties within an ADE that are to be added to AbstractTransportationSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAuxiliaryTrafficArea

Definition:	ADEOfAuxiliaryTrafficArea acts as a hook to define properties within an ADE that are to be added to an AuxiliaryTrafficArea.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAuxiliaryTrafficSpace

Definition:	ADEOfAuxiliaryTrafficSpace acts as a hook to define properties within an ADE that are to be added to an AuxiliaryTrafficSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfClearanceSpace

Definition:	ADEOfClearanceSpace acts as a hook to define properties within an ADE that are to be added to a ClearanceSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfHole

Definition:	ADEOfHole acts as a hook to define properties within an ADE that are to be added to a Hole.
Subclass of:	None
Stereotype:	«DataType»

ADEOfHoleSurface

Definition:	ADEOfHoleSurface acts as a hook to define properties within an ADE that are to be added to a HoleSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfIntersection

Definition:	ADEOfIntersection acts as a hook to define properties within an ADE that are to be added to an Intersection.
Subclass of:	None
Stereotype:	«DataType»

ADEOfMarking

Definition:	ADEOfMarking acts as a hook to define properties within an ADE that are to be added to a Marking.
Subclass of:	None
Stereotype:	«DataType»

ADEOfRailway

Definition:	ADEOfRailway acts as a hook to define properties within an ADE that are to be added to a Railway.
Subclass of:	None
Stereotype:	«DataType»

ADEOfRoad

Definition:	ADEOfRoad acts as a hook to define properties within an ADE that are to be added to a Road.
Subclass of:	None
Stereotype:	«DataType»

ADEOfSection

Definition:	ADEOfSection acts as a hook to define properties within an ADE that are to be added to a Section.
Subclass of:	None
Stereotype:	«DataType»

ADEOfSquare

Definition:	ADEOfSquare acts as a hook to define properties within an ADE that are to be added to a Square.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTrack

Definition:	ADEOfTrack acts as a hook to define properties within an ADE that are to be added to a Track.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTrafficArea

Definition:	ADEOfTrafficArea acts as a hook to define properties within an ADE that are to be added to a TrafficArea.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTrafficSpace

Definition:	ADEOfTrafficSpace acts as a hook to define properties within an ADE that are to be added to a TrafficSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWaterway

Definition:	ADEOfWaterway acts as a hook to define properties within an ADE that are to be added to a Waterway.
Subclass of:	None
Stereotype:	«DataType»

E.10.3. Basic Types

none

E.10.4. Unions

none

E.10.5. Code Lists

AuxiliaryTrafficAreaClassValue

Definition:	AuxiliaryTrafficAreaClassValue is a code list used to further classify an AuxiliaryTrafficArea.
Stereotype:	«CodeList»

AuxiliaryTrafficAreaFunctionValue

Definition:	AuxiliaryTrafficAreaFunctionValue is a code list that enumerates the different purposes of an AuxiliaryTrafficArea.
Stereotype:	«CodeList»

AuxiliaryTrafficAreaUsageValue

Definition:	AuxiliaryTrafficAreaUsageValue is a code list that enumerates the different uses of an AuxiliaryTrafficArea.
Stereotype:	«CodeList»

AuxiliaryTrafficSpaceClassValue

Definition:	AuxiliaryTrafficSpaceClassValue is a code list used to further classify an AuxiliaryTrafficSpace.
Stereotype:	«CodeList»

AuxiliaryTrafficSpaceFunctionValue

Definition:	AuxiliaryTrafficSpaceFunctionValue is a code list that enumerates the different purposes of an AuxiliaryTrafficSpace.
Stereotype:	«CodeList»

AuxiliaryTrafficSpaceUsageValue

Definition:	AuxiliaryTrafficSpaceUsageValue is a code list that enumerates the different uses of an AuxiliaryTrafficSpace.
Stereotype:	«CodeList»

ClearanceSpaceClassValue

Definition:	ClearanceSpaceClassValue is a code list used to further classify a ClearanceSpace.
Stereotype:	«CodeList»

HoleClassValue

Definition:	HoleClassValue is a code list used to further classify a Hole.
Stereotype:	«CodeList»

IntersectionClassValue

Definition:	IntersectionClassValue is a code list used to further classify an Intersection.
Stereotype:	«CodeList»

MarkingClassValue

Definition:	MarkingClassValue is a code list used to further classify a Marking.
Stereotype:	«CodeList»

RailwayClassValue

Definition:	RailwayClassValue is a code list used to further classify a Railway.
Stereotype:	«CodeList»

RailwayFunctionValue

Definition:	RailwayFunctionValue is a code list that enumerates the different purposes of a Railway.
Stereotype:	«CodeList»

RailwayUsageValue

Definition:	RailwayUsageValue is a code list that enumerates the different uses of a Railway.
Stereotype:	«CodeList»

RoadClassValue

Definition:	RoadClassValue is a code list used to further classify a Road.
Stereotype:	«CodeList»

RoadFunctionValue

Definition:	RoadFunctionValue is a code list that enumerates the different purposes of a Road.
Stereotype:	«CodeList»

RoadUsageValue

Definition:	RoadUsageValue is a code list that enumerates the different uses of a Road.
Stereotype:	«CodeList»

SectionClassValue

Definition:	SectionClassValue is a code list used to further classify a Section.
Stereotype:	«CodeList»

SquareClassValue

Definition:	SquareClassValue is a code list used to further classify a Square.
Stereotype:	«CodeList»

SquareFunctionValue

Definition:	SquareFunctionValue is a code list that enumerates the different purposes of a Square.
Stereotype:	«CodeList»

SquareUsageValue

Definition:	SquareUsageValue is a code list that enumerates the different uses of a Square.
Stereotype:	«CodeList»

SurfaceMaterialValue

Definition:	SurfaceMaterialValue is a code list that enumerates the different surface materials.
Stereotype:	«CodeList»

TrackClassValue

Definition:	TrackClassValue is a code list used to further classify a Track.
Stereotype:	«CodeList»

TrackFunctionValue

Definition:	TrackFunctionValue is a code list that enumerates the different purposes of a Track.
Stereotype:	«CodeList»

TrackUsageValue

Definition:	TrackUsageValue is a code list that enumerates the different uses of a Track.
Stereotype:	«CodeList»

TrafficAreaClassValue

Definition:	TrafficAreaClassValue is a code list used to further classify a TrafficArea.
Stereotype:	«CodeList»

TrafficAreaFunctionValue

Definition:	TrafficAreaFunctionValue is a code list that enumerates the different purposes of a TrafficArea.
Stereotype:	«CodeList»

TrafficAreaUsageValue

Definition:	TrafficAreaUsageValue is a code list that enumerates the different uses of a TrafficArea.
Stereotype:	«CodeList»

TrafficSpaceClassValue

Definition:	TrafficSpaceClassValue is a code list used to further classify a TrafficSpace.
Stereotype:	«CodeList»

TrafficSpaceFunctionValue

Definition:	TrafficSpaceFunctionValue is a code list that enumerates the different purposes of a TrafficSpace.
Stereotype:	«CodeList»

TrafficSpaceUsageValue

Definition:	TrafficSpaceUsageValue is a code list that enumerates the different uses of a TrafficSpace.
Stereotype:	«CodeList»

WaterwayClassValue

Definition:	WaterwayClassValue is a code list used to further classify a Waterway.
Stereotype:	«CodeList»

WaterwayFunctionValue

Definition:	WaterwayFunctionValue is a code list that enumerates the different purposes of a Waterway.
Stereotype:	«CodeList»

WaterwayUsageValue

Definition:	WaterwayUsageValue is a code list that enumerates the different uses of a Waterway.
Stereotype:	«CodeList»

E.10.6. Enumerations

GranularityValue

Definition:	GranularityValue enumerates the different levels of granularity in which transportation objects are represented.
StereoType:	<<Enumeration>>

Literal value	Definition
lane	Indicates that the individual lanes of the transportation object are represented.
way	Indicates that the individual (carriage)ways of the transportation object are represented.

TrafficDirectionValue

Definition:	TrafficDirectionValue enumerates the allowed directions of travel of a mobile object.
StereoType:	<<Enumeration>>

Literal value	Definition
forwards	Indicates that traffic flows in the direction of the corresponding linear geometry.
backwards	Indicates that traffic flows in the opposite direction of the corresponding linear geometry.
both	Indicates that traffic flows in both directions.

E.11. Vegetation

Description:	The Vegetation module supports representation of vegetation objects with vegetation-specific thematic classes. CityGML's vegetation model distinguishes between solitary vegetation objects like trees, and vegetation areas which represent biotopes like forests or other plant communities.
Parent Package:	CityGML

Stereotype: «ApplicationSchema»

E.11.1. Classes

AbstractVegetationObject

Definition: AbstractVegetationObject is the abstract superclass for all kinds of vegetation objects.

Subclass of: AbstractOccupiedSpace

Stereotype: «FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractVegetationObject	ADEOfAbstractVegetationObject [0..*]	Augments AbstractVegetationObject with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

PlantCover

Definition: A PlantCover represents a space covered by vegetation.

Subclass of: AbstractVegetationObject

Stereotype: «TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	PlantCoverClassValue [0..1]	Indicates the specific type of the PlantCover.
function	PlantCoverFunctionValue [0..*]	Specifies the intended purposes of the PlantCover.
usage	PlantCoverUsageValue [0..*]	Specifies the actual uses of the PlantCover.
averageHeight	Length [0..1]	Specifies the average height of the PlantCover.
minHeight	Length [0..1]	Specifies the minimum height of the PlantCover.
maxHeight	Length [0..1]	Specifies the maximum height of the PlantCover.
adeOfPlantCover	ADEOfPlantCover [0..*]	Augments the PlantCover with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

SolitaryVegetationObject

Definition:	A SolitaryVegetationObject represents individual vegetation objects, e.g. trees or bushes.
Subclass of:	AbstractVegetationObject
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	SolitaryVegetationObjectClassValue [0..1]	Indicates the specific type of the SolitaryVegetationObject.
function	SolitaryVegetationObjectFunctionValue [0..*]	Specifies the intended purposes of the SolitaryVegetationObject.
usage	SolitaryVegetationObjectUsageValue [0..*]	Specifies the actual uses of the SolitaryVegetationObject.
species	SpeciesValue [0..1]	Indicates the botanical name of the SolitaryVegetationObject.
height	Length [0..1]	Distance between the highest point of the vegetation object and the lowest point of the terrain at the bottom of the object.
trunkDiameter	Length [0..1]	Specifies the diameter of the SolitaryCityObject's trunk.
crownDiameter	Length [0..1]	Specifies the diameter of the SolitaryCityObject's crown.
rootBallDiameter	Length [0..1]	Specifies the diameter of the SolitaryCityObject's root ball.
maxRootBallDepth	Length [0..1]	Specifies the vertical distance between the lowest point of the SolitaryVegetationObject's root ball and the terrain surface.
adeOfSolitaryVegetationObject	ADEOfSolitaryVegetationObject [0..*]	Augments the SolitaryVegetationObject with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.11.2. Data Types

ADEOfAbstractVegetationObject

Definition:	ADEOfAbstractVegetationObject acts as a hook to define properties within an ADE that are to be added to AbstractVegetationObject.
Subclass of:	None
Stereotype:	«DataType»

ADEOfPlantCover

Definition:	ADEOfPlantCover acts as a hook to define properties within an ADE that are to be added to a PlantCover.
Subclass of:	None
Stereotype:	«DataType»

ADEOfSolitaryVegetationObject

Definition:	ADEOfSolitaryVegetationObject acts as a hook to define properties within an ADE that are to be added to a SolitaryVegetationObject.
Subclass of:	None
Stereotype:	«DataType»

E.11.3. Basic Types

none

E.11.4. Unions

none

E.11.5. Code Lists

PlantCoverClassValue

Definition:	PlantCoverClassValue is a code list used to further classify a PlantCover.
Stereotype:	«CodeList»

PlantCoverFunctionValue

Definition:	PlantCoverFunctionValue is a code list that enumerates the different purposes of a PlantCover.
Stereotype:	«CodeList»

PlantCoverUsageValue

Definition:	PlantCoverUsageValue is a code list that enumerates the different uses of a PlantCover.
Stereotype:	«CodeList»

SolitaryVegetationObjectClassValue

Definition:	SolitaryVegetationObjectClassValue is a code list used to further classify a SolitaryVegetationObject.
Stereotype:	«CodeList»

SolitaryVegetationObjectFunctionValue

Definition:	SolitaryVegetationObjectFunctionValue is a code list that enumerates the different purposes of a SolitaryVegetationObject.
Stereotype:	«CodeList»

SolitaryVegetationObjectUsageValue

Definition:	SolitaryVegetationObjectUsageValue is a code list that enumerates the different uses of a SolitaryVegetationObject.
Stereotype:	«CodeList»

SpeciesValue

Definition:	A SpeciesValue is a code list that enumerates the species of a SolitaryVegetationObject.
Stereotype:	«CodeList»

E.11.6. Enumerations

none

E.12. Versioning

Description:	The Versioning module supports representation of multiple versions of CityGML features within a single CityGML model. In addition, also the version transitions and transactions that lead to the different versions can be represented.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.12.1. Classes

Version

Definition:	Version represents a defined state of a city model consisting of the dedicated versions of all city object instances that belong to the respective city model version. Versions can have names, a description and can be labeled with an arbitrary number of user defined tags.
Subclass of:	AbstractVersion
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
versionMember «Version»	AbstractFeatureWithLifespan [*]	Relates to all city objects that are part of the city model version.

Attribute	Value type and multiplicity	Definition
tag	CharacterString [0..*]	Allows for adding keywords to the city model version.
adeOfVersion	ADEOfVersion [0..*]	Augments the Version with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

VersionTransition

Definition:	VersionTransition describes the change of the state of a city model from one version to another. Version transitions can have names, a description and can be further qualified by a type and a reason.
Subclass of:	AbstractVersionTransition
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
from	Version [0..1]	Relates to the predecessor version of the VersionTransition.
transaction	Transaction [*]	Relates to all transactions that have been applied as part of the VersionTransition.
to	Version [0..1]	Relates to the sucessor version of the VersionTransition.

Attribute	Value type and multiplicity	Definition
reason	CharacterString [0..1]	Specifies why the VersionTransition has been carried out.
clonePredecessor	Boolean [1..1]	Indicates whether the set of city object instances belonging to the successor version of the city model is either explicitly enumerated within the successor version object (attribute clonePredecessor=false), or has to be derived from the modifications of the city model provided as a list of transactions on the city object versions contained in the predecessor version (attribute clonePredecessor=true).
type	TransitionTypeValue [0..1]	Indicates the specific type of the VersionTransition.
adeOfVersionTransition	ADEOfVersionTransition [0..*]	Augments the VersionTransition with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.12.2. Data Types

ADEOfVersion

Definition:	ADEOfVersion acts as a hook to define properties within an ADE that are to be added to a Version.
Subclass of:	None
Stereotype:	«DataType»

ADEOfVersionTransition

Definition:	ADEOfVersionTransition acts as a hook to define properties within an ADE that are to be added to a VersionTransition.
Subclass of:	None
Stereotype:	«DataType»

Transaction

Definition:	Transaction represents a modification of the city model by the creation, termination, or replacement of a specific city object. While the creation of a city object also marks its first object version, the termination marks the end of existence of a real world object and, hence, also terminates the final version of a city object. The replacement of a city object means that a specific version of it is replaced by a new version.
Subclass of:	None
Stereotype:	«DataType»

Role name	Target class and multiplicity	Definition
newFeature «Version»	AbstractFeatureWithLifespan [0..1]	Relates to the version of the city object subsequent to the Transaction.
oldFeature «Version»	AbstractFeatureWithLifespan [0..1]	Relates to the version of the city object prior to the Transaction.

Attribute	Value type and multiplicity	Definition
type	TransactionTypeValue [1..1]	Indicates the specific type of the Transaction.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.12.3. Basic Types

none

E.12.4. Unions

none

E.12.5. Code Lists

none

E.12.6. Enumerations

TransactionTypeValue

Definition:	TransactionTypeValue enumerates the three possible types of transactions: insert, delete, or replace.
StereoType:	<<Enumeration>>

Literal value	Definition
insert	Indicates that the feature referenced from the Transaction via the "newFeature" association has been newly created; the association "oldFeature" is empty in this case.
delete	Indicates that the feature referenced from the Transaction via the "oldFeature" association ceases to exist; the association "newFeature" is empty in this case.
replace	Indicates that the feature referenced from the Transaction via the "oldFeature" association has been replaced by the feature referenced via the "newFeature" association.

TransitionTypeValue

Definition:	TransitionTypeValue enumerates the different kinds of version transitions. “planned” and “fork” should be used in cases when from one city model version multiple successor versions are being created. “realized” and “merge” should be used when different city model versions are converging into a common successor version.
StereoType:	<<Enumeration>>

Literal value	Definition
planned	Indicates that the successor version of the city model represents a planning state for a possible future of the city.
realized	Indicates that the predecessor version is the chosen one from a number of possible planning versions.
historicalSuccession	Indicates that the successor version reflects updates on the city model over time (historical timeline). It shall only be used for at most one version transition outgoing from a city model version.
fork	Indicates other reasons to create alternative city model versions, for example, when different parties are updating parts of the city model or to reflect the results of different simulation runs.
merge	Indicates other reasons to converge multiple versions back into a common city model version.

E.13. WaterBody

Description:	The WaterBody module supports representation of the thematic aspects and 3D geometry of rivers, canals, lakes, and basins. It does, however, not inherit any hydrological or other dynamic aspects of fluid flow.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.13.1. Classes

AbstractWaterBoundarySurface

Definition:	AbstractWaterBoundarySurface is the abstract superclass for all kinds of thematic surfaces bounding a water body.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractWaterBoundarySurface	ADEOfAbstractWaterBoundarySurface [0..*]	Augments AbstractWaterBoundarySurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

WaterBody

Definition:	A WaterBody represents significant and permanent or semi-permanent accumulations of surface water, usually covering a part of the Earth.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
boundary	AbstractWaterBoundarySurface [*]	

Attribute	Value type and multiplicity	Definition
class	WaterBodyClassValue [0..1]	Indicates the specific type of the WaterBody.
function	WaterBodyFunctionValue [0..*]	Specifies the intended purposes of the WaterBody.
usage	WaterBodyUsageValue [0..*]	Specifies the actual uses of the WaterBody.
adeOfWaterBody	ADEOfWaterBody [0..*]	Augments the WaterBody with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

WaterGroundSurface

Definition:	A WaterGroundSurface represents the exterior boundary surface of the submerged bottom of a water body.
Subclass of:	AbstractWaterBoundarySurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfWaterGroundSurface	ADEOfWaterGroundSurface [0..*]	Augments the WaterGroundSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

WaterSurface

Definition:	A WaterSurface represents the upper exterior interface between a water body and the atmosphere.
Subclass of:	AbstractWaterBoundarySurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
waterLevel	WaterLevelValue [0..1]	Specifies the level of the WaterSurface.
adeOfWaterSurface	ADEOfWaterSurface [0..*]	Augments the WaterSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.13.2. Data Types

ADEOfAbstractWaterBoundarySurface

Definition:	ADEOfAbstractWaterBoundarySurface acts as a hook to define properties within an ADE that are to be added to AbstractWaterBoundarySurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWaterBody

Definition:	ADEOfWaterBody acts as a hook to define properties within an ADE that are to be added to a WaterBody.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWaterGroundSurface

Definition:	ADEOfWaterGroundSurface acts as a hook to define properties within an ADE that are to be added to a WaterGroundSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWaterSurface

Definition:	ADEOfWaterSurface acts as a hook to define properties within an ADE that are to be added to a WaterSurface.
Subclass of:	None
Stereotype:	«DataType»

E.13.3. Basic Types

none

E.13.4. Unions

none

E.13.5. Code Lists

WaterBodyClassValue

Definition:	WaterBodyClassValue is a code list used to further classify a WaterBody.
Stereotype:	«CodeList»

WaterBodyFunctionValue

Definition:	WaterBodyFunctionValue is a code list that enumerates the different purposes of a WaterBody.
Stereotype:	«CodeList»

WaterBodyUsageValue

Definition:	WaterBodyUsageValue is a code list that enumerates the different uses of a WaterBody.
Stereotype:	«CodeList»

WaterLevelValue

Definition:	WaterLevelValue is a code list that enumerates the different levels of a water surface.
Stereotype:	«CodeList»

E.13.6. Enumerations

none

E.14. Construction

Description:	The Construction module supports representation of key elements of different types of constructions. These key elements include construction surfaces (e.g floor and ceiling), windows and doors, constructive elements (e.g. beams and slabs), installations, and furniture.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.14.1. Classes

AbstractConstruction

Definition:	AbstractConstruction is the abstract superclass for objects that are manufactured by humans from construction materials, are connected to earth, and are intended to be permanent. A connection with the ground also exists when the construction rests by its own weight on the ground or is moveable limited on stationary rails or if the construction is intended to be used mainly stationary.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the construction. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
conditionOfConstruction	ConditionOfConstructionValue [0..1]	Indicates the life-cycle status of the construction. [cf. INSPIRE]
dateOfConstruction	Date [0..1]	Indicates the date at which the construction was completed.
dateOfDemolition	Date [0..1]	Indicates the date at which the construction was demolished.
constructionEvent	ConstructionEvent [0..*]	Describes specific events in the life-time of the construction.
elevation	Elevation [0..*]	Specifies qualified elevations of the construction in relation to a well-defined surface which is commonly taken as origin (e.g. geoid or water level). [cf. INSPIRE]
height	Height [0..*]	Specifies qualified heights of the construction above ground or below ground. [cf. INSPIRE]
occupancy	Occupancy [0..*]	Provides qualified information on the residency of persons, animals, or other moveable objects in the construction.
adeOfAbstractConstruction	ADEOfAbstractConstruction [0..*]	Augments AbstractConstruction with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractConstructionSurface

Definition:	AbstractConstructionSurface is the abstract superclass for different kinds of surfaces that bound a construction.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
fillingSurface	AbstractFillingSurface [*]	Relates to the surfaces that seal the openings of the construction surface.

Attribute	Value type and multiplicity	Definition
adeOfAbstractConstructionSurface	ADEOfAbstractConstructionSurface [0..*]	Augments AbstractConstructionSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractConstructiveElement

Definition:	AbstractConstructiveElement is the abstract superclass for the representation of volumetric elements of a construction. Examples are walls, beams, slabs.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the constructive element. This relation is inherited from the Core module.
filling	AbstractFillingElement [*]	Relates to the elements that fill the opening of the constructive element.

Attribute	Value type and multiplicity	Definition
isStructuralElement	Boolean [0..1]	Indicates whether the constructive element is essential from a structural point of view. A structural element cannot be omitted without collapsing of the construction. Examples are pylons and anchorages of bridges.
adeOfAbstractConstructiveElement	ADEOfAbstractConstructiveElement [0..*]	Augments AbstractConstructiveElement with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractFillingElement

Definition:	AbstractFillingElement is the abstract superclass for different kinds of elements that fill the openings of a construction.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractFillingElement	ADEOfAbstractFillingElement [0..*]	Augments AbstractFillingElement with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractFillingSurface

Definition:	AbstractFillingSurface is the abstract superclass for different kinds of surfaces that seal openings filled by filling elements.
Subclass of:	AbstractThematicSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractFillingSurface	ADEOfAbstractFillingSurface [0..*]	Augments AbstractFillingSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractFurniture

Definition:	AbstractFurniture is the abstract superclass for the representation of furniture objects of a construction.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfAbstractFurniture	ADEOfAbstractFurniture [0..*]	Augments AbstractFurniture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractInstallation

Definition:	AbstractInstallation is the abstract superclass for the representation of installation objects of a construction.
Subclass of:	AbstractOccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the installation. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
relationToConstruction	RelationToConstruction [0..1]	Indicates whether the installation is located inside and/or outside of the construction.
adeOfAbstractInstallation	ADEOfAbstractInstallation [0..*]	Augments AbstractInstallation with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

CeilingSurface

Definition:	A CeilingSurface is a surface that represents the interior ceiling of a construction. An example is the ceiling of a room.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfCeilingSurface	ADEOfCeilingSurface [0..*]	Augments the CeilingSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Door

Definition:	A Door is a construction for closing an opening intended primarily for access or egress or both. [cf. ISO 6707-1]
Subclass of:	AbstractFillingElement
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
address	Address [*]	Relates to the addresses that are assigned to the Door.
boundary	DoorSurface [*]	Relates to the door surfaces that bound the Door. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	DoorClassValue [0..1]	Indicates the specific type of the Door.
function	DoorFunctionValue [0..*]	Specifies the intended purposes of the Door.
usage	DoorUsageValue [0..*]	Specifies the actual uses of the Door.
adeOfDoor	ADEOfDoor [0..*]	Augments the Door with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

DoorSurface

Definition:	A DoorSurface is either a boundary surface of a Door feature or a surface that seals an opening filled by a door.
Subclass of:	AbstractFillingSurface
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
address	Address [*]	Relates to the addresses that are assigned to the DoorSurface.

Attribute	Value type and multiplicity	Definition
adeOfDoorSurface	ADEOfDoorSurface [0..*]	Augments the DoorSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

FloorSurface

Definition:	A FloorSurface is surface that represents the interior floor of a construction. An example is the floor of a room.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfFloorSurface	ADEOfFloorSurface [0..*]	Augments the FloorSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

GroundSurface

Definition:	A GroundSurface is a surface that represents the ground plate of a construction. The polygon defining the ground plate is congruent with the footprint of the construction.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfGroundSurface	ADEOfGroundSurface [0..*]	Augments the GroundSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

InteriorWallSurface

Definition:	An InteriorWallSurface is a surface that is visible from inside a construction. An example is the wall of a room.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfInteriorWallSurface	ADEOfInteriorWallSurface [0..*]	Augments the InteriorWallSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

OtherConstruction

Definition:	An OtherConstruction is a construction that is not covered by any of the other subclasses of AbstractConstruction.
Subclass of:	AbstractConstruction
Stereotype:	«TopLevelFeatureType»

Attribute	Value type and multiplicity	Definition
class	OtherConstructionClassValue [0..1]	Indicates the specific type of the OtherConstruction.
function	OtherConstructionFunctionValue [0..*]	Specifies the intended purposes of the OtherConstruction.
usage	OtherConstructionUsageValue [0..*]	Specifies the actual uses of the OtherConstruction.
adeOfOtherConstruction	ADEOfOtherConstruction [0..*]	Augments the OtherConstruction with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

OuterCeilingSurface

Definition:	An OuterCeilingSurface is a surface that belongs to the outer building shell with the orientation pointing downwards. An example is the ceiling of a loggia.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfOuterCeilingSurface	ADEOfOuterCeilingSurface [0..*]	Augments the OuterCeilingSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

OuterFloorSurface

Definition:	An OuterFloorSurface is a surface that belongs to the outer construction shell with the orientation pointing upwards. An example is the floor of a loggia.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfOuterFloorSurface	ADEOfOuterFloorSurface [0..*]	Augments the OuterFloorSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

RoofSurface

Definition:	A RoofSurface is a surface that delimits major roof parts of a construction.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfRoofSurface	ADEOfRoofSurface [0..*]	Augments the RoofSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

WallSurface

Definition:	A WallSurface is a surface that is part of the building facade visible from the outside.
Subclass of:	AbstractConstructionSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfWallSurface	ADEOfWallSurface [0..*]	Augments the WallSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Window

Definition:	A Window is a construction for closing an opening in a wall or roof, primarily intended to admit light and/or provide ventilation. [cf. ISO 6707-1]
Subclass of:	AbstractFillingElement
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	WindowSurface [*]	Relates to the window surfaces that bound the Window. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	WindowClassValue [0..1]	Indicates the specific type of the Window.
function	WindowFunctionValue [0..*]	Specifies the intended purposes of the Window.
usage	WindowUsageValue [0..*]	Specifies the actual uses of the Window.
adeOfWindow	ADEOfWindow [0..*]	Augments the Window with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

WindowSurface

Definition:	A WindowSurface is either a boundary surface of a Window feature or a surface that seals an opening filled by a window.
Subclass of:	AbstractFillingSurface
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfWindowSurface	ADEOfWindowSurface [0..*]	Augments the WindowSurface with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.14.2. Data Types

ADEOfAbstractConstruction

Definition:	ADEOfAbstractConstruction acts as a hook to define properties within an ADE that are to be added to AbstractConstruction.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractConstructionSurface

Definition:	ADEOfAbstractConstructionSurface acts as a hook to define properties within an ADE that are to be added to AbstractConstructionSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractConstructiveElement

Definition:	ADEOfAbstractConstructiveElement acts as a hook to define properties within an ADE that are to be added to AbstractConstructiveElement.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractFillingElement

Definition:	ADEOfAbstractFillingElement acts as a hook to define properties within an ADE that are to be added to AbstractFillingElement.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractFillingSurface

Definition:	ADEOfAbstractFillingSurface acts as a hook to define properties within an ADE that are to be added to AbstractFillingSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractFurniture

Definition:	ADEOfAbstractFurniture acts as a hook to define properties within an ADE that are to be added to AbstractFurniture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractInstallation

Definition:	ADEOfAbstractInstallation acts as a hook to define properties within an ADE that are to be added to AbstractInstallation.
Subclass of:	None
Stereotype:	«DataType»

ADEOfCeilingSurface

Definition:	ADEOfCeilingSurface acts as a hook to define properties within an ADE that are to be added to a CeilingSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfDoor

Definition:	ADEOfDoor acts as a hook to define properties within an ADE that are to be added to a Door.
Subclass of:	None
Stereotype:	«DataType»

ADEOfDoorSurface

Definition:	ADEOfDoorSurface acts as a hook to define properties within an ADE that are to be added to a DoorSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfFloorSurface

Definition:	ADEOfFloorSurface acts as a hook to define properties within an ADE that are to be added to a FloorSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfGroundSurface

Definition:	ADEOfGroundSurface acts as a hook to define properties within an ADE that are to be added to a GroundSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfInteriorWallSurface

Definition:	ADEOfInteriorWallSurface acts as a hook to define properties within an ADE that are to be added to an InteriorWallSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfOtherConstruction

Definition:	ADEOfOtherConstruction acts as a hook to define properties within an ADE that are to be added to an OtherConstruction.
Subclass of:	None
Stereotype:	«DataType»

ADEOfOuterCeilingSurface

Definition:	ADEOfOuterCeilingSurface acts as a hook to define properties within an ADE that are to be added to an OuterCeilingSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfOuterFloorSurface

Definition:	ADEOfOuterFloorSurface acts as a hook to define properties within an ADE that are to be added to an OuterFloorSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfRoofSurface

Definition:	ADEOfRoofSurface acts as a hook to define properties within an ADE that are to be added to a RoofSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWallSurface

Definition:	ADEOfWallSurface acts as a hook to define properties within an ADE that are to be added to a WallSurface.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWindow

Definition:	ADEOfWindow acts as a hook to define properties within an ADE that are to be added to a Window.
Subclass of:	None
Stereotype:	«DataType»

ADEOfWindowSurface

Definition:	ADEOfWindowSurface acts as a hook to define properties within an ADE that are to be added to a WindowSurface.
Subclass of:	None
Stereotype:	«DataType»

ConstructionEvent

Definition:	A ConstructionEvent is a data type used to describe a specific event that is associated with a construction. Examples are the issuing of a building permit or the renovation of a building.
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
event	EventValue [1..1]	Indicates the specific event type.
dateOfEvent	Date [1..1]	Specifies the date at which the event took or will take place.
description	CharacterString [0..1]	Provides additional information on the event.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Elevation

Definition:	Elevation is a data type that includes the elevation value itself and information on how this elevation was measured. [cf. INSPIRE]
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
elevationReference	ElevationReferenceValue [1..1]	Specifies the level from which the elevation was measured. [cf. INSPIRE]
elevationValue	DirectPosition [1..1]	Specifies the value of the elevation. [cf. INSPIRE]

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Height

Definition:	Height represents a vertical distance (measured or estimated) between a low reference and a high reference. [cf. INSPIRE]
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
highReference	ElevationReferenceValue [1..1]	Indicates the high point used to calculate the value of the height. [cf. INSPIRE]
lowReference	ElevationReferenceValue [1..1]	Indicates the low point used to calculate the value of the height. [cf. INSPIRE]
status	HeightStatusValue [1..1]	Indicates the way the height has been captured. [cf. INSPIRE]
value	Length [1..1]	Specifies the value of the height above or below ground. [cf. INSPIRE]

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.14.3. Basic Types

none

E.14.4. Unions

none

E.14.5. Code Lists

DoorClassValue

Definition:	DoorClassValue is a code list used to further classify a Door.
Stereotype:	«CodeList»

DoorFunctionValue

Definition:	DoorFunctionValue is a code list that enumerates the different purposes of a Door.
Stereotype:	«CodeList»

DoorUsageValue

Definition:	DoorUsageValue is a code list that enumerates the different uses of a Door.
Stereotype:	«CodeList»

ElevationReferenceValue

Definition:	ElevationReferenceValue is a code list that enumerates the different elevation reference levels used to measure construction heights.
Stereotype:	«CodeList»

EventValue

Definition:	EventValue is a code list that enumerates the different events of a construction.
Stereotype:	«CodeList»

OtherConstructionClassValue

Definition:	OtherConstructionClassValue is a code list used to further classify an OtherConstruction.
Stereotype:	«CodeList»

OtherConstructionFunctionValue

Definition:	OtherConstructionFunctionValue is a code list that enumerates the different purposes of an OtherConstruction.
Stereotype:	«CodeList»

OtherConstructionUsageValue

Definition:	OtherConstructionUsageValue is a code list that enumerates the different uses of an OtherConstruction.
Stereotype:	«CodeList»

WindowClassValue

Definition:	WindowClassValue is a code list used to further classify a Window.
Stereotype:	«CodeList»

WindowFunctionValue

Definition:	WindowFunctionValue is a code list that enumerates the different purposes of a Window.
Stereotype:	«CodeList»

WindowUsageValue

Definition:	WindowUsageValue is a code list that enumerates the different uses of a Window.
Stereotype:	«CodeList»

E.14.6. Enumerations

ConditionOfConstructionValue

Definition:	ConditionOfConstructionValue enumerates different conditions of a construction. [cf. INSPIRE]
StereoType:	<<Enumeration>>

Literal value	Definition
declined	Indicates that the construction cannot be used under normal conditions, though its main elements (walls, roof) are still present. [cf. INSPIRE]
demolished	Indicates that the construction has been demolished. There are no more visible remains. [cf. INSPIRE]
functional	Indicates that the construction is functional. [cf. INSPIRE]
projected	Indicates that the construction is being designed. Construction works have not yet started. [cf. INSPIRE]
ruin	Indicates that the construction has been partly demolished and some main elements (roof, walls) have been destroyed. There are some visible remains of the construction. [cf. INSPIRE]
underConstruction	Indicates that the construction is under construction and not yet functional. This applies only to the initial construction works of the construction and not to maintenance work. [cf. INSPIRE]

HeightStatusValue

Definition:	HeightStatusValue enumerates the different methods used to capture a height. [cf. INSPIRE]
StereoType:	<<Enumeration>>

Literal value	Definition
estimated	Indicates that the height has been estimated and not measured. [cf. INSPIRE]
measured	Indicates that the height has been (directly or indirectly) measured. [cf. INSPIRE]

RelationToConstruction

Definition:	RelationToConstruction is an enumeration used to describe whether an installation is positioned inside and/or outside of a construction.
StereoType:	<<Enumeration>>

Literal value	Definition
inside	Indicates that the installation is positioned inside of the construction.
outside	Indicates that the installation is positioned outside of the construction.
bothInsideAndOutside	Indicates that the installation is positioned inside as well as outside of the construction.

E.15. Bridge

Description:	The Bridge module supports representation of thematic and spatial aspects of bridges, bridge parts, bridge installations, and interior bridge structures.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.15.1. Classes

AbstractBridge

Definition:	AbstractBridge is an abstract superclass representing the common attributes and associations of the classes Bridge and BridgePart.
Subclass of:	AbstractConstruction
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
bridgeConstructiveElement	BridgeConstructiveElement [*]	Relates the constructive elements to the Bridge or BridgePart.
bridgeInstallation	BridgeInstallation [*]	Relates the installation objects to the Bridge or BridgePart.
bridgeFurniture	BridgeFurniture [*]	Relates the furniture objects to the Bridge or BridgePart.
bridgeRoom	BridgeRoom [*]	Relates the rooms to the Bridge or BridgePart.
address	Address [*]	Relates the addresses to the Bridge or BridgePart.

Attribute	Value type and multiplicity	Definition
class	BridgeClassValue [0..1]	Indicates the specific type of the Bridge or BridgePart.
function	BridgeFunctionValue [0..*]	Specifies the intended purposes of the Bridge or BridgePart.
usage	BridgeUsageValue [0..*]	Specifies the actual uses of the Bridge or BridgePart.
isMovable	Boolean [0..1]	Indicates whether the Bridge or BridgePart can be moved to allow for watercraft to pass.
adeOfAbstractBridge	ADEOfAbstractBridge [0..*]	Augments AbstractBridge with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Bridge

Definition:	A Bridge represents a structure that affords the passage of pedestrians, animals, vehicles, and service(s) above obstacles or between two points at a height above ground. [cf. ISO 6707-1]
Subclass of:	AbstractBridge
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
bridgePart	BridgePart [*]	Relates the bridge parts to the Bridge.

Attribute	Value type and multiplicity	Definition
adeOfBridge	ADEOfBridge [0..*]	Augments the Bridge with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BridgeConstructiveElement

Definition:	A BridgeConstructiveElement is an element of a bridge which is essential from a structural point of view. Examples are pylons, anchorages, slabs, beams.
Subclass of:	AbstractConstructiveElement
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	BridgeConstructiveElementClassValue [0..1]	Indicates the specific type of the BridgeConstructiveElement.
function	BridgeConstructiveElementFunctionValue [0..*]	Specifies the intended purposes of the BridgeConstructiveElement.
usage	BridgeConstructiveElementUsageValue [0..*]	Specifies the actual uses of the BridgeConstructiveElement.
adeOfBridgeConstructiveElement	ADEOfBridgeConstructiveElement [0..*]	Augments the BridgeConstructiveElement with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BridgeFurniture

Definition:	A BridgeFurniture is an equipment for occupant use, usually not fixed to the bridge. [cf. ISO 6707-1]
Subclass of:	AbstractFurniture
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	BridgeFurnitureClassValue [0..1]	Indicates the specific type of the BridgeFurniture.
function	BridgeFurnitureFunctionValue [0..*]	Specifies the intended purposes of the BridgeFurniture.
usage	BridgeFurnitureUsageValue [0..*]	Specifies the actual uses of the BridgeFurniture.
adeOfBridgeFurniture	ADEOfBridgeFurniture [0..*]	Augments the BridgeFurniture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BridgeInstallation

Definition:	A BridgeInstallation is a permanent part of a Bridge (inside and/or outside) which does not have the significance of a BridgePart. In contrast to BridgeConstructiveElements, a BridgeInstallation is not essential from a structural point of view. Examples are stairs, antennas or railways.
Subclass of:	AbstractInstallation
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	BridgeInstallationClassValue [0..1]	Indicates the specific type of the BridgeInstallation.
function	BridgeInstallationFunctionValue [0..*]	Specifies the intended purposes of the BridgeInstallation.
usage	BridgeInstallationUsageValue [0..*]	Specifies the actual uses of the BridgeInstallation.
adeOfBridgeInstallation	ADEOfBridgeInstallation [0..*]	Augments the BridgeInstallation with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BridgePart

Definition:	A BridgePart is a physical or functional subdivision of a Bridge. It would be considered a Bridge, if it were not part of a collection of other BridgeParts.
Subclass of:	AbstractBridge
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfBridgePart	ADEOfBridgePart [0..*]	Augments the BridgePart with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BridgeRoom

Definition:	A BridgeRoom is a space within a Bridge or BridgePart intended for human occupancy (e.g. a place of work or recreation) and/or containment (storage) of animals or things. A BridgeRoom is bounded physically and/or virtually (e.g. by ClosureSurfaces or GenericSurfaces).
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
bridgeInstallation	BridgeInstallation [*]	Relates to the installation objects to the BridgeRoom.
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the BridgeRoom. This relation is inherited from the Core module.
bridgeFurniture	BridgeFurniture [*]	Relates the furniture objects to the BridgeRoom.

Attribute	Value type and multiplicity	Definition
class	BridgeRoomClassValue [0..1]	Indicates the specific type of the BridgeRoom.
function	BridgeRoomFunctionValue [0..*]	Specifies the intended purposes of the BridgeRoom.
usage	BridgeRoomUsageValue [0..*]	Specifies the actual uses of the BridgeRoom.
adeOfBridgeRoom	ADEOfBridgeRoom [0..*]	Augments the BridgeRoom with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.15.2. Data Types

ADEOfAbstractBridge

Definition:	ADEOfAbstractBridge acts as a hook to define properties within an ADE that are to be added to AbstractBridge.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBridge

Definition:	ADEOfBridge acts as a hook to define properties within an ADE that are to be added to a Bridge.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBridgeConstructiveElement

Definition:	ADEOfBridgeConstructiveElement acts as a hook to define properties within an ADE that are to be added to a BridgeConstructiveElement.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBridgeFurniture

Definition:	ADEOfBridgeFurniture acts as a hook to define properties within an ADE that are to be added to a BridgeFurniture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBridgeInstallation

Definition:	ADEOfBridgeInstallation acts as a hook to define properties within an ADE that are to be added to a BridgeInstallation.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBridgePart

Definition:	ADEOfBridgePart acts as a hook to define properties within an ADE that are to be added to a BridgePart.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBridgeRoom

Definition:	ADEOfBridgeRoom acts as a hook to define properties within an ADE that are to be added to a BridgeRoom.
Subclass of:	None
Stereotype:	«DataType»

E.15.3. Basic Types

none

E.15.4. Unions

none

E.15.5. Code Lists

BridgeClassValue

Definition:	BridgeClassValue is a code list used to further classify a Bridge.
Stereotype:	«CodeList»

BridgeConstructiveElementClassValue

Definition:	BridgeConstructiveElementClassValue is a code list used to further classify a BridgeConstructiveElement.
Stereotype:	«CodeList»

BridgeConstructiveElementFunctionValue

Definition:	BridgeConstructiveElementFunctionValue is a code list that enumerates the different purposes of a BridgeConstructiveElement.
Stereotype:	«CodeList»

BridgeConstructiveElementUsageValue

Definition:	BridgeConstructiveElementUsageValue is a code list that enumerates the different uses of a BridgeConstructiveElement.
Stereotype:	«CodeList»

BridgeFunctionValue

Definition:	BridgeFunctionValue is a code list that enumerates the different purposes of a Bridge.
Stereotype:	«CodeList»

BridgeFurnitureClassValue

Definition:	BridgeFurnitureClassValue is a code list used to further classify a BridgeFurniture.
Stereotype:	«CodeList»

BridgeFurnitureFunctionValue

Definition:	BridgeFurnitureFunctionValue is a code list that enumerates the different purposes of a BridgeFurniture.
Stereotype:	«CodeList»

BridgeFurnitureUsageValue

Definition:	BridgeFurnitureUsageValue is a code list that enumerates the different uses of a BridgeFurniture.
Stereotype:	«CodeList»

BridgeInstallationClassValue

Definition:	BridgeInstallationClassValue is a code list used to further classify a BridgeInstallation.
Stereotype:	«CodeList»

BridgeInstallationFunctionValue

Definition:	BridgeInstallationFunctionValue is a code list that enumerates the different purposes of a BridgeInstallation.
Stereotype:	«CodeList»

BridgeInstallationUsageValue

Definition:	BridgeInstallationUsageValue is a code list that enumerates the different uses of a BridgeInstallation.
Stereotype:	«CodeList»

BridgeRoomClassValue

Definition:	BridgeRoomClassValue is a code list used to further classify a BridgeRoom.
Stereotype:	«CodeList»

BridgeRoomFunctionValue

Definition:	BridgeRoomFunctionValue is a code list that enumerates the different purposes of a BridgeRoom.
Stereotype:	«CodeList»

BridgeRoomUsageValue

Definition:	BridgeRoomUsageValue is a code list that enumerates the different uses of a BridgeRoom.
Stereotype:	«CodeList»

BridgeUsageValue

Definition:	BridgeUsageValue is a code list that enumerates the different uses of a Bridge.
Stereotype:	«CodeList»

E.15.6. Enumerations

none

E.16. Building

Description:	The Building module supports representation of thematic and spatial aspects of buildings, building parts, building installations, building subdivisions, and interior building structures.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.16.1. Classes

AbstractBuilding

Definition:	AbstractBuilding is an abstract superclass representing the common attributes and associations of the classes Building and BuildingPart.
Subclass of:	AbstractConstruction
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
buildingFurniture	BuildingFurniture [*]	Relates the furniture objects to the Building or BuildingPart.
buildingRoom	BuildingRoom [*]	Relates the rooms to the Building or BuildingPart.
buildingInstallation	BuildingInstallation [*]	Relates the installation objects to the Building or BuildingPart.
buildingSubdivision	AbstractBuildingSubdivision [*]	Relates the logical subdivisions to the Building or BuildingPart.
buildingConstructiveElement	BuildingConstructiveElement [*]	Relates the constructive elements to the Building or BuildingPart.
address	Address [*]	Relates the addresses to the Building or BuildingPart.

Attribute	Value type and multiplicity	Definition
class	BuildingClassValue [0..1]	Indicates the specific type of the Building or BuildingPart.
function	BuildingFunctionValue [0..*]	Specifies the intended purposes of the Building or BuildingPart.
usage	BuildingUsageValue [0..*]	Specifies the actual uses of the Building or BuildingPart.
roofType	RoofTypeValue [0..1]	Indicates the shape of the roof of the Building or BuildingPart.
storeysAboveGround	Integer [0..1]	Indicates the number of storeys positioned above ground level.
storeysBelowGround	Integer [0..1]	Indicates the number of storeys positioned below ground level.
storeyHeightsAboveGround	MeasureOrNilReasonList [0..1]	Lists the heights of each storey above ground. The first value in the list denotes the height of the storey closest to the ground level, the last value denotes the height furthest away.
storeyHeightsBelowGround	MeasureOrNilReasonList [0..1]	Lists the height of each storey below ground. The first value in the list denotes the height of the storey closest to the ground level, the last value denotes the height furthest away.
adeOfAbstractBuilding	ADEOfAbstractBuilding [0..*]	Augments AbstractBuilding with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

AbstractBuildingSubdivision

Definition:	AbstractBuildingSubdivision is the abstract superclass for different kinds of logical building subdivisions.
Subclass of:	AbstractLogicalSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
buildingRoom	BuildingRoom [*]	Relates the rooms to the building subdivision.
buildingFurniture	BuildingFurniture [*]	Relates the furniture objects to the building subdivision.
buildingConstructiveElement	BuildingConstructiveElement [*]	Relates the constructive elements to the building subdivision.
buildingInstallation	BuildingInstallation [*]	Relates the installation objects to the building subdivision.

Attribute	Value type and multiplicity	Definition
class	BuildingSubdivisionClassValue [0..1]	Indicates the specific type of the building subdivision.
function	BuildingSubdivisionFunctionValue [0..*]	Specifies the intended purposes of the building subdivision.
usage	BuildingSubdivisionUsageValue [0..*]	Specifies the actual uses of the building subdivision.
elevation	Elevation [0..*]	Specifies qualified elevations of the building subdivision in relation to a well-defined surface which is commonly taken as origin (e.g. geoid or water level). [cf. INSPIRE]
sortKey	Real [0..1]	Defines an order among the objects that belong to the building subdivision. An example is the sorting of storeys.
adeOfAbstractBuildingSubdivision	ADEOfAbstractBuildingSubdivision [0..*]	Augments AbstractBuildingSubdivision with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Building

Definition:	A Building is a free-standing, self-supporting construction that is roofed, usually walled, and can be entered by humans and is normally designed to stand permanently in one place. It is intended for human occupancy (e.g. a place of work or recreation), habitation and/or shelter of humans, animals or things.
Subclass of:	AbstractBuilding
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
buildingPart	BuildingPart [*]	Relates the building parts to the Building.

Attribute	Value type and multiplicity	Definition
adeOfBuilding	ADEOfBuilding [0..*]	Augments the Building with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BuildingConstructiveElement

Definition:	A BuildingConstructiveElement is an element of a Building which is essential from a structural point of view. Examples are walls, slabs, staircases, beams.
Subclass of:	AbstractConstructiveElement
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	BuildingConstructiveElementClassValue [0..1]	Indicates the specific type of the BuildingConstructiveElement.
function	BuildingConstructiveElementFunctionValue [0..*]	Specifies the intended purposes of the BuildingConstructiveElement.
usage	BuildingConstructiveElementUsageValue [0..*]	Specifies the actual uses of the BuildingConstructiveElement.
adeOfBuildingConstructiveElement	ADEOfBuildingConstructiveElement [0..*]	Augments the BuildingConstructiveElement with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BuildingFurniture

Definition:	A BuildingFurniture is an equipment for occupant use, usually not fixed to the building. [cf. ISO 6707-1]
Subclass of:	AbstractFurniture
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	BuildingFurnitureClassValue [0..1]	Indicates the specific type of the BuildingFurniture.
function	BuildingFurnitureFunctionValue [0..*]	Specifies the intended purposes of the BuildingFurniture.
usage	BuildingFurnitureUsageValue [0..*]	Specifies the actual uses of the BuildingFurniture.
adeOfBuildingFurniture	ADEOfBuildingFurniture [0..*]	Augments the BuildingFurniture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BuildingInstallation

Definition:	A BuildingInstallation is a permanent part of a Building (inside and/or outside) which has not the significance of a BuildingPart. Examples are stairs, antennas, balconies or small roofs.
Subclass of:	AbstractInstallation
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	BuildingInstallationClassValue [0..1]	Indicates the specific type of the BuildingInstallation.
function	BuildingInstallationFunctionValue [0..*]	Specifies the intended purposes of the BuildingInstallation.
usage	BuildingInstallationUsageValue [0..*]	Specifies the actual uses of the BuildingInstallation.
adeOfBuildingInstallation	ADEOfBuildingInstallation [0..*]	Augments the BuildingInstallation with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BuildingPart

Definition:	A BuildingPart is a physical or functional subdivision of a Building. It would be considered a Building, if it were not part of a collection of other BuildingParts.
Subclass of:	AbstractBuilding
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfBuildingPart	ADEOfBuildingPart [0..*]	Augments the BuildingPart with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BuildingRoom

Definition:	A BuildingRoom is a space within a Building or BuildingPart intended for human occupancy (e.g. a place of work or recreation) and/or containment of animals or things. A BuildingRoom is bounded physically and/or virtually (e.g. by ClosureSurfaces or GenericSurfaces).
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
buildingInstallation	BuildingInstallation [*]	Relates the installation objects to the BuildingRoom.
buildingFurniture	BuildingFurniture [*]	Relates the furniture objects to the BuildingRoom.
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the BuildingRoom. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	BuildingRoomClassValue [0..1]	Indicates the specific type of the BuildingRoom.
function	BuildingRoomFunctionValue [0..*]	Specifies the intended purposes of the BuildingRoom.
usage	BuildingRoomUsageValue [0..*]	Specifies the actual uses of the BuildingRoom.
roomHeight	RoomHeight [0..*]	Specifies qualified heights of the BuildingRoom.
adeOfBuildingRoom	ADEOfBuildingRoom [0..*]	Augments the BuildingRoom with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

BuildingUnit

Definition:	A BuildingUnit is a logical subdivision of a Building. BuildingUnits are formed according to some homogeneous property like function, ownership, management, or accessability. They may be separately sold, rented out, inherited, managed, etc.
Subclass of:	AbstractBuildingSubdivision
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
storey	Storey [*]	Relates to the storeys on which the BuildingUnit is located.
address	Address [*]	Relates to the addresses that are assigned to the BuildingUnit.

Attribute	Value type and multiplicity	Definition
adeOfBuildingUnit	ADEOfBuildingUnit [0..*]	Augments the BuildingUnit with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Storey

Definition:	A Storey is typically a horizontal section of a Building. Storeys are not always defined according to the building structure, but can also be defined according to logical considerations.
Subclass of:	AbstractBuildingSubdivision
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the Storey. This relation is inherited from the Core module.
buildingUnit	BuildingUnit [*]	Relates to the building units that belong to the Storey.

Attribute	Value type and multiplicity	Definition
adeOfStorey	ADEOfStorey [0..*]	Augments the Storey with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.16.2. Data Types

ADEOfAbstractBuilding

Definition:	ADEOfAbstractBuilding acts as a hook to define properties within an ADE that are to be added to AbstractBuilding.
Subclass of:	None
Stereotype:	«DataType»

ADEOfAbstractBuildingSubdivision

Definition:	ADEOfAbstractBuildingSubdivision acts as a hook to define properties within an ADE that are to be added to AbstractBuildingSubdivision.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuilding

Definition:	ADEOfBuilding acts as a hook to define properties within an ADE that are to be added to a Building.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuildingConstructiveElement

Definition:	ADEOfBuildingConstructiveElement acts as a hook to define properties within an ADE that are to be added to a BuildingConstructiveElement.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuildingFurniture

Definition:	ADEOfBuildingFurniture acts as a hook to define properties within an ADE that are to be added to a BuildingFurniture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuildingInstallation

Definition:	ADEOfBuildingInstallation acts as a hook to define properties within an ADE that are to be added to a BuildingInstallation.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuildingPart

Definition:	ADEOfBuildingPart acts as a hook to define properties within an ADE that are to be added to a BuildingPart.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuildingRoom

Definition:	ADEOfBuildingRoom acts as a hook to define properties within an ADE that are to be added to a BuildingRoom.
Subclass of:	None
Stereotype:	«DataType»

ADEOfBuildingUnit

Definition:	ADEOfBuildingUnit acts as a hook to define properties within an ADE that are to be added to a BuildingUnit.
Subclass of:	None
Stereotype:	«DataType»

ADEOfStorey

Definition:	ADEOfStorey acts as a hook to define properties within an ADE that are to be added to a Storey.
Subclass of:	None
Stereotype:	«DataType»

RoomHeight

Definition:	The RoomHeight represents a vertical distance (measured or estimated) between a low reference and a high reference. [cf. INSPIRE]
Subclass of:	None
Stereotype:	«DataType»

Attribute	Value type and multiplicity	Definition
highReference	RoomElevationReferenceValue [1..1]	Indicates the high point used to calculate the value of the room height.
lowReference	RoomElevationReferenceValue [1..1]	Indicates the low point used to calculate the value of the room height.
status	HeightStatusValue [1..1]	Indicates the way the room height has been captured.
value	Length [1..1]	Specifies the value of the room height.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.16.3. Basic Types

none

E.16.4. Unions

none

E.16.5. Code Lists

BuildingClassValue

Definition:	BuildingClassValue is a code list used to further classify a Building.
Stereotype:	«CodeList»

BuildingConstructiveElementClassValue

Definition:	BuildingConstructiveElementClassValue is a code list used to further classify a BuildingConstructiveElement.
Stereotype:	«CodeList»

BuildingConstructiveElementFunctionValue

Definition:	BuildingConstructiveElementFunctionValue is a code list that enumerates the different purposes of a BuildingConstructiveElement.
Stereotype:	«CodeList»

BuildingConstructiveElementUsageValue

Definition:	BuildingConstructiveElementUsageValue is a code list that enumerates the different uses of a BuildingConstructiveElement.
Stereotype:	«CodeList»

BuildingFunctionValue

Definition:	BuildingFunctionValue is a code list that enumerates the different purposes of a Building.
Stereotype:	«CodeList»

BuildingFurnitureClassValue

Definition:	BuildingFurnitureClassValue is a code list used to further classify a BuildingFurniture.
Stereotype:	«CodeList»

BuildingFurnitureFunctionValue

Definition:	BuildingFurnitureFunctionValue is a code list that enumerates the different purposes of a BuildingFurniture.
Stereotype:	«CodeList»

BuildingFurnitureUsageValue

Definition:	BuildingFurnitureUsageValue is a code list that enumerates the different uses of a BuildingFurniture.
Stereotype:	«CodeList»

BuildingInstallationClassValue

Definition:	BuildingInstallationClassValue is a code list used to further classify a BuildingInstallation.
Stereotype:	«CodeList»

BuildingInstallationFunctionValue

Definition:	BuildingInstallationFunctionValue is a code list that enumerates the different purposes of a BuildingInstallation.
Stereotype:	«CodeList»

BuildingInstallationUsageValue

Definition:	BuildingInstallationUsageValue is a code list that enumerates the different uses of a BuildingInstallation.
Stereotype:	«CodeList»

BuildingRoomClassValue

Definition:	BuildingRoomClassValue is a code list used to further classify a BuildingRoom.
Stereotype:	«CodeList»

BuildingRoomFunctionValue

Definition:	BuildingRoomFunctionValue is a code list that enumerates the different purposes of a BuildingRoom.
Stereotype:	«CodeList»

BuildingRoomUsageValue

Definition:	BuildingRoomUsageValue is a code list that enumerates the different uses of a BuildingRoom.
Stereotype:	«CodeList»

BuildingSubdivisionClassValue

Definition:	BuildingSubdivisionClassValue is a code list used to further classify a BuildingSubdivision.
Stereotype:	«CodeList»

BuildingSubdivisionFunctionValue

Definition:	BuildingSubdivisionFunctionValue is a code list that enumerates the different purposes of a BuildingSubdivision.
Stereotype:	«CodeList»

BuildingSubdivisionUsageValue

Definition:	BuildingSubdivisionUsageValue is a code list that enumerates the different uses of a BuildingSubdivision.
Stereotype:	«CodeList»

BuildingUsageValue

Definition:	BuildingUsageValue is a code list that enumerates the different uses of a Building.
Stereotype:	«CodeList»

RoofTypeValue

Definition:	RoofTypeValue is a code list that enumerates different roof types.
Stereotype:	«CodeList»

RoomElevationReferenceValue

Definition:	RoomElevationReferenceValue is a code list that enumerates the different elevation reference levels used to measure room heights.
Stereotype:	«CodeList»

E.16.6. Enumerations

none

E.17. Tunnel

Description:	The Tunnel module supports representation of thematic and spatial aspects of tunnels, tunnel parts, tunnel installations, and interior tunnel structures.
Parent Package:	CityGML
Stereotype:	«ApplicationSchema»

E.17.1. Classes

AbstractTunnel

Definition:	AbstractTunnel is an abstract superclass representing the common attributes and associations of the classes Tunnel and TunnelPart.
Subclass of:	AbstractConstruction
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
hollowSpace	HollowSpace [*]	Relates the hollow spaces to the Tunnel or TunnelPart.
tunnelConstructiveElement	TunnelConstructiveElement [*]	Relates the constructive elements to the Tunnel or TunnelPart.
tunnelInstallation	TunnelInstallation [*]	Relates the installation objects to the Tunnel or TunnelPart.
tunnelFurniture	TunnelFurniture [*]	Relates the furniture objects to the Tunnel or TunnelPart.

Attribute	Value type and multiplicity	Definition
class	TunnelClassValue [0..1]	Indicates the specific type of the Tunnel or TunnelPart.
function	TunnelFunctionValue [0..*]	Specifies the intended purposes of the Tunnel or TunnelPart.
usage	TunnelUsageValue [0..*]	Specifies the actual uses of the Tunnel or TunnelPart.
adeOfAbstractTunnel	ADEOfAbstractTunnel [0..*]	Augments AbstractTunnel with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

HollowSpace

Definition:	A HollowSpace is a space within a Tunnel or TunnelPart intended for certain functions (e.g. transport or passage ways, service rooms, emergency shelters). A HollowSpace is bounded physically and/or virtually (e.g. by ClosureSurfaces or GenericSurfaces).
Subclass of:	AbstractUnoccupiedSpace
Stereotype:	«FeatureType»

Role name	Target class and multiplicity	Definition
tunnelInstallation	TunnelInstallation [*]	Relates the installation objects to the HollowSpace.
tunnelFurniture	TunnelFurniture [*]	Relates the furniture objects to the HollowSpace.
boundary	AbstractThematicSurface [*]	Relates to the surfaces that bound the HollowSpace. This relation is inherited from the Core module.

Attribute	Value type and multiplicity	Definition
class	HollowSpaceClassValue [0..1]	Indicates the specific type of the HollowSpace.
function	HollowSpaceFunctionValue [0..*]	Specifies the intended purposes of the HollowSpace.
usage	HollowSpaceUsageValue [0..*]	Specifies the actual uses of the HollowSpace.
adeOfHollowSpace	ADEOfHollowSpace [0..*]	Augments the HollowSpace with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

Tunnel

Definition:	A Tunnel represents a horizontal or sloping enclosed passage way of a certain length, mainly underground or underwater. [cf. ISO 6707-1]
Subclass of:	AbstractTunnel
Stereotype:	«TopLevelFeatureType»

Role name	Target class and multiplicity	Definition
tunnelPart	TunnelPart [*]	Relates the tunnel parts to the Tunnel.

Attribute	Value type and multiplicity	Definition
adeOfTunnel	ADEOfTunnel [0..*]	Augments the Tunnel with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TunnelConstructiveElement

Definition:	A TunnelConstructiveElement is an element of a Tunnel which is essential from a structural point of view. Examples are walls, slabs, beams.
Subclass of:	AbstractConstructiveElement
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	TunnelConstructiveElementClassValue [0..1]	Indicates the specific type of the TunnelConstructiveElement.
function	TunnelConstructiveElementFunctionValue [0..*]	Specifies the intended purposes of the TunnelConstructiveElement.
usage	TunnelConstructiveElementUsageValue [0..*]	Specifies the actual uses of the TunnelConstructiveElement.
adeOfTunnelConstructiveElement	ADEOfTunnelConstructiveElement [0..*]	Augments the TunnelConstructiveElement with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TunnelFurniture

Definition:	A TunnelFurniture is an equipment for occupant use, usually not fixed to the tunnel. [cf. ISO 6707-1]
Subclass of:	AbstractFurniture
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	TunnelFurnitureClassValue [0..1]	Indicates the specific type of the TunnelFurniture.
function	TunnelFurnitureFunctionValue [0..*]	Specifies the intended purposes of the TunnelFurniture.
usage	TunnelFurnitureUsageValue [0..*]	Specifies the actual uses of the TunnelFurniture.
adeOfTunnelFurniture	ADEOfTunnelFurniture [0..*]	Augments the TunnelFurniture with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TunnelInstallation

Definition:	A TunnelInstallation is a permanent part of a Tunnel (inside and/or outside) which does not have the significance of a TunnelPart. In contrast to TunnelConstructiveElement, a TunnelInstallation is not essential from a structural point of view. Examples are stairs, antennas or railings.
Subclass of:	AbstractInstallation
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
class	TunnelInstallationClassValue [0..1]	Indicates the specific type of the TunnelInstallation.
function	TunnelInstallationFunctionValue [0..*]	Specifies the intended purposes of the TunnelInstallation.
usage	TunnelInstallationUsageValue [0..*]	Specifies the actual uses of the TunnelInstallation.
adeOfTunnelInstallation	ADEOfTunnelInstallation [0..*]	Augments the TunnelInstallation with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

TunnelPart

Definition:	A TunnelPart is a physical or functional subdivision of a Tunnel. It would be considered a Tunnel, if it were not part of a collection of other TunnelParts.
Subclass of:	AbstractTunnel
Stereotype:	«FeatureType»

Attribute	Value type and multiplicity	Definition
adeOfTunnelPart	ADEOfTunnelPart [0..*]	Augments the TunnelPart with properties defined in an ADE.

Note: Unless otherwise specified, all attributes and role names have the stereotype «Property».

E.17.2. Data Types

ADEOfAbstractTunnel

Definition:	ADEOfAbstractTunnel acts as a hook to define properties within an ADE that are to be added to AbstractTunnel.
Subclass of:	None
Stereotype:	«DataType»

ADEOfHollowSpace

Definition:	ADEOfHollowSpace acts as a hook to define properties within an ADE that are to be added to a HollowSpace.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTunnel

Definition:	ADEOfTunnel acts as a hook to define properties within an ADE that are to be added to a Tunnel.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTunnelConstructiveElement

Definition:	ADEOfTunnelConstructiveElement acts as a hook to define properties within an ADE that are to be added to a TunnelConstructiveElement.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTunnelFurniture

Definition:	ADEOfTunnelFurniture acts as a hook to define properties within an ADE that are to be added to a TunnelFurniture.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTunnelInstallation

Definition:	ADEOfTunnelInstallation acts as a hook to define properties within an ADE that are to be added to a TunnelInstallation.
Subclass of:	None
Stereotype:	«DataType»

ADEOfTunnelPart

Definition:	ADEOfTunnelPart acts as a hook to define properties within an ADE that are to be added to a TunnelPart.
Subclass of:	None
Stereotype:	«DataType»

E.17.3. Basic Types

none

E.17.4. Unions

none

E.17.5. Code Lists

HollowSpaceClassValue

Definition:	HollowSpaceClassValue is a code list used to further classify a HollowSpace.
Stereotype:	«CodeList»

HollowSpaceFunctionValue

Definition:	HollowSpaceFunctionValue is a code list that enumerates the different purposes of a HollowSpace.
Stereotype:	«CodeList»

HollowSpaceUsageValue

Definition:	HollowSpaceUsageValue is a code list that enumerates the different uses of a HollowSpace.
Stereotype:	«CodeList»

TunnelClassValue

Definition:	TunnelClassValue is a code list used to further classify a Tunnel.
Stereotype:	«CodeList»

TunnelConstructiveElementClassValue

Definition:	TunnelConstructiveElementClassValue is a code list used to further classify a TunnelConstructiveElement.
Stereotype:	«CodeList»

TunnelConstructiveElementFunctionValue

Definition:	TunnelConstructiveElementFunctionValue is a code list that enumerates the different purposes of a TunnelConstructiveElement.
Stereotype:	«CodeList»

TunnelConstructiveElementUsageValue

Definition:	TunnelConstructiveElementUsageValue is a code list that enumerates the different uses of a TunnelConstructiveElement.
Stereotype:	«CodeList»

TunnelFunctionValue

Definition:	TunnelFunctionValue is a code list that enumerates the different purposes of a Tunnel.
Stereotype:	«CodeList»

TunnelFurnitureClassValue

Definition:	TunnelFurnitureClassValue is a code list used to further classify a TunnelFurniture.
Stereotype:	«CodeList»

TunnelFurnitureFunctionValue

Definition:	TunnelFurnitureFunctionValue is a code list that enumerates the different purposes of a TunnelFurniture.
Stereotype:	«CodeList»

TunnelFurnitureUsageValue

Definition:	TunnelFurnitureUsageValue is a code list that enumerates the different uses of a TunnelFurniture.
Stereotype:	«CodeList»

TunnelInstallationClassValue

Definition:	TunnelInstallationClassValue is a code list used to further classify a TunnelInstallation.
Stereotype:	«CodeList»

TunnelInstallationFunctionValue

Definition:	TunnelInstallationFunctionValue is a code list that enumerates the different purposes of a TunnelInstallation.
Stereotype:	«CodeList»

TunnelInstallationUsageValue	
Definition:	TunnelInstallationUsageValue is a code list that enumerates the different uses of a TunnelInstallation.
Stereotype:	«CodeList»

TunnelUsageValue	
Definition:	TunnelUsageValue is a code list that enumerates the different uses of a Tunnel.
Stereotype:	«CodeList»

E.17.6. Enumerations

none

Annex F: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version

Annex G: Glossary

conformance test class
set of conformance test modules that must be applied to receive a single certificate of conformance
[OGC 08-131r3, definition 4.4]

feature
abstraction of real world phenomena
[ISO 19101-1:2014, definition 4.1.11]

feature attribute
characteristic of a feature
[ISO 19101-1:2014, definition 4.1.12]

feature type
class of features having common characteristics
[ISO 19156:2011, definition 4.7]

measurement
set of operations having the object of determining the value of a quantity
[ISO 19101-2:2018, definition 3.21] / [VIM:1993, 2.1]

model
abstraction of some aspects of reality
[ISO 19109:2015, definition 4.15]

observation
act of measuring or otherwise determining the value of a property
[ISO 19156:2011, definition 4.11]

observation procedure
method, algorithm or instrument, or system of these, which may be used in making an observation
[ISO 19156:2011, 4.12]

observation result
estimate of the value of a property determined through a known observation procedure
[ISO 19156:2011, 4.14]

property
facet or attribute of an object referenced by a name.
[ISO 19143:2010, definition 4.21]

requirements class
aggregate of all requirement modules that must all be satisfied to satisfy a conformance test class
[OGC 08-131r3, definition 4.19]

schema
formal description of a model
[ISO 19101-1:2014, definition 4.1.34]

sensor
type of observation procedure that provides the estimated value of an observed property at its output
[OGC 08-094r1, definition 4.5]

Standardization Target
TBD

timeseries
sequence of data values which are ordered in time
[OGC 15-043r3]

universe of discourse
view of the real or hypothetical world that includes everything of interest
[ISO 19101-1:2014, definition 4.1.38]

version
Particular variation of a spatial object
[INSPIRE Glossary]

G.1. ISO Concepts

The following concepts from the ISO TC211 Harmonized UML model are referenced by the CityGML Conceptual UML model but do not play a major role in its' definition. They are provided here to support a more complete understanding of the model.

Area
The measure of the physical extent of any topologically 2-D geometric object. Usually measured in "square" units of length.
[ISO 19103:2015]

Boolean
boolean is the mathematical datatype associated with two-valued logic
[ISO 19103:2015]

CC_CoordinateOperation
mathematical operation on coordinates that transforms or converts coordinates to another coordinate reference system.
[ISO 19111:2019]

Character
symbol from a standard character-set.
[ISO 19103:2015]

CharacterString
Characterstring is a family of datatypes which represent strings of symbols from standard character-sets.
[ISO 19103:2015]

CRS

Coordinate reference system which is usually single but may be compound.
[ISO 19111:2019]

CV_DiscreteCoverage
A subclass of CV_Coverage that returns a single record of values for any direct position within a single geometric object in its spatiotemporal domain.
[ISO 19123:2005]

CV_DomainObject
[ISO 19123:2005]

CV_GridPointValuePair
[ISO 19123:2005]

CV_GridValuesMatrix
The geometry represented by the various offset vectors is in the image plane of the grid.
[ISO 19123:2005]

CV_ReferenceableGrid
[ISO 19123:2005]

Date
Date gives values for year, month and day. Representation of Date is specified in ISO 8601. Principles for date and time are further discussed in ISO 19108.
[ISO 19103:2015]

DateTime
A DateTime is a combination of a date and a time types. Representation of DateTime is specified in ISO 8601. Principles for date and time are further discussed in ISO 19108.
[ISO 19103:2015]

Distance
Used as a type for returning distances and possibly lengths.
[ISO 19103:2015]

Engineering CRS
A contextually local coordinate reference system which can be divided into two broad categories:

1. earth-fixed systems applied to engineering activities on or near the surface of the earth;
2. CRSs on moving platforms such as road vehicles, vessels, aircraft or spacecraft.
   [ISO 19111:2019]

Generic Name
Generic Name is the abstract class for all names in a NameSpace. Each instance of a GenericName is either a LocalName or a ScopedName.
[ISO 19103:2015]

Geometry
[ISO 19107:2003]

GM_CompositePoint
[ISO 19107:2003]

GM_CompositeSolid
set of geometric solids adjoining one another along common boundary geometric surfaces
[ISO 19107:2003]

GM_GenericSurface
GM_Surface and GM_SurfacePatch both represent sections of surface geometry, and therefore share a number of operation signatures. These are defined in the interface class GM_GenericSurface.
[ISO 19107:2003]

GM_LineString
consists of sequence of line segments, each having a parameterization like the one for GM_LineSegment
[ISO 19107:2003]

GM_MultiPrimitive
[ISO 19107:2003]

GM_OrientableSurface
a surface and an orientation inherited from GM_OrientablePrimitive. If the orientation is "+", then the GM_OrientableSurface is a GM_Surface. If the orientation is "-", then the GM_OrientableSurface is a reference to a GM_Surface with an upNormal that reverses the direction for this GM_OrientableSurface, the sense of "the top of the surface".
[ISO 19107:2003]

GM_PolyhedralSurface
a GM_Surface composed of polygon surfaces (GM_Polygon) connected along their common boundary curves.
[ISO 19107:2003]

GM_Position
a union type consisting of either a DirectPosition or of a reference to a GM_Point from which a DirectPosition shall be obtained.
[ISO 19107:2003]

GM_Primitive
The abstract root class of the geometric primitives. Its main purpose is to define the basic "boundary" operation that ties the primitives in each dimension together.
[ISO 19107:2003]

Integer
An exact integer value, with no fractional part.
[ISO 19103:2015]

Internet of Things
The network of physical objects--“things”--that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.

Wikipedia

IO_IdentifiedObjectBase
[ISO 19103:2015]

Length
The measure of distance as an integral, i.e. the limit of an infinite sum of distances between points on a curve.
[ISO 19103:2015]

Measure
The result from performing the act or process of ascertaining the extent, dimensions, or quantity of some entity.
[ISO 19103:2015]

Number
The base type for all number data, giving the basic algebraic operations.
[ISO 19103:2015]

Point
GM_Point is the basic data type for a geometric object consisting of one and only one point.
[ISO 19107:2003]

Real
The common binary Real finite implementation using base 2.
[ISO 19103:2015]

RS_ReferenceSystem
Description of a spatial and temporal reference system used by a dataset.
[ISO 19111:2019]

Scoped Name
ScopedName is a composite of a LocalName for locating another NameSpace and a GenericName valid in that NameSpace. ScopedName contains a LocalName as head and a GenericName, which might be a LocalName or a ScopedName, as tail.
[ISO 19103:2015]

Solid
GM_Solid, a subclass of GM_Primitive, is the basis for 3-dimensional geometry. The extent of a solid is defined by the boundary surfaces.
[ISO 19107:2003]

Time
Time is the designation of an instant on a selected time scale, astronomical or atomic. It is used in the sense of time of day.
[ISO 19103:2015]

TM_Duration
[ISO 19108:2006]

TM_TemporalPosition
The position of a TM_Instant relative to a TM_ReferenceSystem.
[ISO 19108:2006]

Unit of Measure
Any of the systems devised to measure some physical quantity such distance or area or a system devised to measure such things as the passage of time.
[ISO 19103:2015]

URI
Uniform Resource Identifier (URI), is a compact string of characters used to identify or name a resource
[ISO 19103:2015]

Volume
Volume is the measure of the physical space of any 3-D geometric object.
[ISO 19103:2015]

G.2. Abbreviated Terms

- 2D Two Dimensional
- 3D Three Dimensional
- AEC Architecture, Engineering, Construction
- ALKIS German National Standard for Cadastral Information
- ATKIS German National Standard for Topographic and Cartographic Information
- BIM Building Information Modeling
- B-Rep Boundary Representation
- bSI buildingSMART International
- CAD Computer Aided Design
- COLLADA Collaborative Design Activity
- CSG Constructive Solid Geometry
- DTM Digital Terrain Model
- DXF Drawing Exchange Format
- EuroSDR European Spatial Data Research Organisation
- ESRI Environmental Systems Research Institute
- FM Facility Management
- GDF Geographic Data Files
- GDI-DE Spatial Data Infrastructure Germany (Geodateninfrastruktur Deutschland)
- GDI NRW Geodata Infrastructure North-Rhine Westphalia
- GML Geography Markup Language

- IAI International Alliance for Interoperability (now buildingSMART International (bSI))
- IETF Internet Engineering Task Force
- IFC Industry Foundation Classes
- IoT Internet of Things
- ISO International Organization for Standardisation
- ISO/TC211 ISO Technical Committee 211
- LOD Levels of Detail
- MQTT
- NBIMS National Building Information Model Standard
- OASIS Organisation for the Advancement of Structured Information Standards
- OGC Open Geospatial Consortium
- OSCRE Open Standards Consortium for Real Estate
- SIG 3D Special Interest Group 3D of the GDI-DE
- TIC Terrain Intersection Curve
- TIN Triangulated Irregular Network
- UML Unified Modeling Language
- URI Uniform Resource Identifier
- VRML Virtual Reality Modeling Language
- W3C World Wide Web Consortium
- W3DS OGC Web 3D Service
- WFS OGC Web Feature Service
- X3D Open Standards XML-enabled 3D file format of the Web 3D Consortium
- XML Extensible Markup Language
- xAL OASIS extensible Address Language

Annex H: Bibliography

- Open Geospatial Consortium: **The Specification Model — A Standard for Modular specifications**, OGC 08-131
- Agugiaro, G., Benner, J., Cipriano, P., Nouvel, R., 2018: **The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations**. Open Geospatial Data, Software and Standards, Vol. 3. https://doi.org/10.1186/s40965-018-0042-y
- Becker, T., Nagel, C., Kolbe, T. H., 2011: **Integrated 3D Modeling of Multi-utility Networks and their Interdependencies for Critical Infrastructure Analysis**. In: T. H. Kolbe, G. König, C. Nagel (Eds.): Advances in 3D Geoinformation Sciences. LNG&C, Springer, Berlin. https://doi.org/10.1007/978-3-642-12670-3_1
- Beil, C., Kolbe, T. H., 2017: **CityGML and the streets of New York - A proposal for detailed street space modelling**. In: Proceedings of the 12th International 3D GeoInfo Conference 2017, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W5, ISPRS. http://doi.org/10.5194/isprs-annals-IV-4-W5-9-2017
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015: **Applications of 3D City Models: State of the Art Review**. ISPRS International Journal of Geo-Information, 4(4). https://doi.org/10.3390/ijgi4042842
- Biljecki, F., Kumar, K., Nagel, C., 2018: **CityGML Application Domain Extension (ADE): overview of developments**. Open Geospatial Data, Software and Standards, 3(1). https://doi.org/10.1186/s40965-018-0055-6
- Billen, R., Zaki, C. E., Servières, M., Moreau, G., Hallot, P., 2012: **Developing an ontology of space: Application to 3D city modeling**. In: Leduc, T., Moreau, G., Billen, R. (eds): Usage, usability, and utility of 3D city models — European COST Action TU0801, EDP Sciences, Nantes, Vol. 02007. https://hal.archives-ouvertes.fr/hal-01521445
- Chaturvedi, K., Smyth, C. S., Gesquière, G., Kutzner, T., Kolbe, T. H., 2015: **Managing versions and history within semantic 3D city models for the next generation of CityGML**. In: Selected papers from the 10th International 3DGeoInfo Conference 2015 in Kuala Lumpur, Malaysia, Springer LNG&C, Berlin. https://doi.org/10.1007/978-3-319-25691-7_11
- Chaturvedi, K., Kolbe, T. H., 2016: **Integrating Dynamic Data and Sensors with Semantic 3D City Models in the context of Smart Cities**. In: Proceedings of the 11th International 3D Geoinfo Conference, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-2/W1, ISPRS. https://doi.org/10.5194/isprs-annals-IV-2-W1-31-2016
- Chaturvedi, K., Kolbe, T. H., 2017: **Future City Pilot 1 Engineering Report**, Open Geospatial Consortium. OGC Doc. 19-098
- Chaturvedi, K., Kolbe, T. H., 2019: **A Requirement Analysis on Extending Semantic 3D City Models for Supporting Time-dependent Properties**. In: Proceedings of the 4th International Conference on Smart Data and Smart Cities, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W9, ISPRS. https://doi.org/10.5194/isprs-annals-IV-4-W9-19-2019
- Elfes, A., 1989: **Using occupancy grids for mobile robot perception and navigation**. Computer 22(6):46–57. https://doi.org/10.1109/2.30720

- Foley, J., van Dam, A., Feiner, S., Hughes, J., 2002: **Computer Graphics: Principles and Practice**. 2nd ed., Addison Wesley
- Gröger, G., Plümer, L., 2012: **CityGML – Interoperable semantic 3D city models**. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 71, July 2012. https://dx.doi.org/10.1016/j.isprsjprs.2012.04.004
- Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H., 2012: **OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0**, Open Geospatial Consortium. OGC Doc. 12-019
- Jensen, Christian S. and Dyreson, Curtis E.: **The Consensus Glossary of Temporal Database Concepts**. February 1998 Version. In: Temporal Databases: Research and Practice [online]. Springer Berlin Heidelberg, 1998. p. 367–405. Lecture Notes in Computer Science. Available from: 10.1007/BFb0053710
- Jensen, Christian S. and Snodgrass, Richard T., eds.: **TR-90, Temporal Database Entries for the Springer Encyclopedia of Database Systems**. Technical Report. TimeCenter, 22 May 2008. Available from: http://timecenter.cs.aau.dk/TimeCenterPublications/TR-90.pdf
- Johnson, Tom: **Bitemporal Data**. Elsevier, 2014. ISBN 978-0-12-408067-6. Available from: 10.1016/C2012-0-06609-4
- Kaden, R., Clemen, C., 2017: **Applying Geodetic Coordinate Reference Systems within Building Information Modeling (BIM)**. In: Proceedings of the FIG Working Week 2017, Helsinki, Finland. https://www.fig.net/resources/proceedings/fig_proceedings/fig2017/papers/ts06h/TS06H_kaden_clemen_8967.pdf
- Kolbe, T. H., Gröger, G., 2003: **Towards unified 3D city models**. In: Proceedings of the Joint ISPRS Commission IV Workshop on Challenges in Geospatial Analysis, Integration and Visualization II, Stuttgart, Germany. https://mediatum.ub.tum.de/doc/1145769/
- Kolbe, T. H., 2009: **Representing and Exchanging 3D City Models with CityGML**. In: J. Lee, S. Zlatanova (Eds.), 3D Geo-Information Sciences, Selected Papers of the 3rd International Workshop on 3D Geo-Information in Seoul, Korea. Springer, Berlin. https://doi.org/10.1007/978-3-540-87395-2_2
- Konde, A., Tauscher, H., Biljecki, F., Crawford, J., 2018: **Floor plans in CityGML**. In: Proceedings of the 13th 3D GeoInfo Conference 2018, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W6, 25–32, ISPRS. https://doi.org/10.5194/isprs-annals-IV-4-W6-25-2018
- Kutzner, T., Hijazi, I., Kolbe, T. H., 2018: **Semantic Modelling of 3D Multi-utility Networks for Urban Analyses and Simulations – The CityGML Utility Network ADE**. International Journal of 3-D Information Modeling (IJ3DIM) 7(2), 1-34. https://dx.doi.org/10.4018/IJ3DIM.2018040101
- Kutzner, T., Chaturvedi, K. & Kolbe, T. H., 2020: **CityGML 3.0: New Functions Open Up New Applications**. PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 88, 43–61. https://doi.org/10.1007/s41064-020-00095-z
- Labetski, A., van Gerwen, S., Tamminga, G., Ledoux, H., Stoter, J., 2018: **A proposal for an improved transportation model in CityGML**. In: Proceedings of the 13th 3D GeoInfo Conference 2018, ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLII-4/W10, 89–96. https://doi.org/10.5194/isprs-archives-XLII-4-W10-89-2018
- Liu, Ling and Özsu, M. Tamer, eds.: **Encyclopedia of Database Systems**. New York, NY :

Springer New York, 2018. ISBN 978-1-4614-8266-6. Available from: 10.1007/978-1-4614-8265-9

- Löwner, M.-O., Gröger, G., Benner, J., Biljecki, F., Nagel, C., 2016: **Proposal for a new LOD and multi-representation concept for CityGML**. In: Proceedings of the 11th 3D Geoinfo Conference 2016, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-2/W1, 3–12. https://doi.org/10.5194/isprs-annals-IV-2-W1-3-2016
- Nouvel, R., Bahu, J. M., Kaden, R., Kaempf, J., Cipriano, P., Lauster, M., Haefele, K.-H., Munoz, E., Tournaire, O, Casper, E., 2015: **Development of the CityGML Application Domain Extension Energy for Urban Energy Simulation**. In: Proceedings of Building Simulation 2015 - 14th Conference of the International Building Performance Simulation Association, IBPSA, 559-564. http://www.ibpsa.org/proceedings/BS2015/p2863.pdf
- Smith, B., Varzi, A. C., 2000: **Fiat and Bona Fide Boundaries**. Philosophy and Phenomenological Research, Vol. 60, No. 2, 401-420. https://doi.org/10.2307/2653492
- Snodgrass, Richard T: **Developing time-oriented database applications in SQL**. San Francisco, California : Morgan Kaufmann Publishers, July 1999. ISBN 1-55860-436-7. Available from: http://www.cs.arizona.edu/rts/tdbbook.pdf[http://www.cs.arizona.edu/rts/tdbbook.pdf]
- Stadler, A., Kolbe, T. H., 2007: **Spatio-semantic Coherence in the Integration of 3D City Models**. In: Proceedings of the 5th International ISPRS Symposium on Spatial Data Quality ISSDQ 2007 in Enschede. http://www.isprs.org/proceedings/XXXVI/2-C43/Session1/paper_Stadler.pdf
- Vretanos, P. A. 2010: **OpenGIS Web Feature Service 2.0 Interface Standard**, Open Geospatial Consortium. OGC Doc. 09-025r1
- OASIS MQTT Technical Committee: **MQTT Version 5.0 Standard**, OASIS, March 7, 2019, Available from OASIS.
- Reed, C., Belayneh T.: **OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package Format Specification**, Open Geospatial Consortium, Available from OGC Doc. 17-014r7
- [[3dtiles_citation, OGC 3D Tiles]]Cozzi, P., Lilley, S., Getz, G. **OGC 3D Tiles Specification 1.0** Open Geospatial Consortium, Available from OGC Doc. 18-053r2
- Burggraf, D.: **OGC KML 2.3**, Open Geospatial Consortium, Available from OGC Doc. 12-007r2
- Bröring, A., Stasch, C., Echterhoff, J.: **OGC® Sensor Observation Service Interface Standard**, Open Geospatial Consortium, Available from OGC Doc. 12-006
- Liang, S., Huang, C., Khalafbeigi, T.: **OGC SensorThings API Part 1: Sensing**, Open Geospatial Consortium, Available from OGC Doc. 15-078r6
- [[3dps_citation, OGC 3D Portrayal Service]]Hagedorn, B., Thum, S., Reitz, T., Coors, V., Gutbell, R.: **OGC® 3D Portrayal Service 1.0**, Open Geospatial Consortium, Available from OGC Doc. 15-001r3.
- Bhatia, S.,Cozzi, P., Knyazev, A., Parisi, T.: **The GL Transmission Format (glTF)**, The Khronos Group, Available from https://www.khronos.org/gltf.