Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2020-01-06>

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

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

Version: 0.3

Category: OGC® Implementation Specification

Editor: Charles Heazel

OGC City Geography Markup Language (CityGML) Conceptual Model 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:	Conceptual Modele
Document stage:	Draft
Document language:	English

Table of Contents

i. Abstract

CityGML is an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is an application schema for the Geography Markup Language version 3.2.1 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211.

The aim of the development of CityGML is to reach a common definition of the basic entities, attributes, and relations of a 3D city model. This is especially important with respect to the cost-effective sustainable maintenance of 3D city models, allowing the reuse of the same data in different application fields.

ii. Keywords

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

ogcdoc, OGC document, <tags separated by commas>

iii. Preface

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.

There are significant changes between CityGML version 2.0.0 and CityGML version 1.0.0 (OGC document no. 08-007r1):

- New thematic modules for the representation of tunnels and bridges;
- Additional boundary surfaces for the semantic classification of the outer shell of buildings and building parts (OuterCeilingSurface and OuterFloorSurface);
- LOD0 representation (footprint and roof egde representations) for buildings and building parts;
- Additional attributes denoting a city object’s location with respect to the surrounding terrain and water surface (relativeToTerrain and relativeToWater);
- Additional generic attributes for measured values and attribute sets; and
- Redesign of the CityGML code list mechanism (enumerative attributes are now of type gml:CodeType which facilitates to provide additional code lists enumerating their possible attribute values).

Migration of existing CityGML 1.0 instances to valid 2.0 instances only requires changing the CityGML namespace and schema location values in the document to the actual 2.0 values.

iv. Submitting organizations

This is the official CityGML logo. For current news on CityGML and information about ongoing projects and fields of research in the area of CityGML see http://www.citygml.org and http://www.citygmlwiki.org

NOTE	PNG interlace method not supported for PDF generation. Find a more compatible OGC logo for "image::images/OGC_Logo.png[]"

OGC work on CityGML is discussed and coordinated by the OGC 3D Information Management (3DIM) Working Group.CityGMLwas initally implemented and evaluated as part of the OGC Web Services Testbed, Phase 4 (OWS-4) in the CAD/GIS/BIM thread.

Version 2.0 of this standards document was prepared by the OGC CityGML Standards Working Group (SWG). Future discussion and development will be led by the 3DIM Working Group.

For further information see http://www.opengeospatial.org/projects/groups/3dimwg

CityGML also continues to be developed by the members of the Special Interest Group 3D (SIG 3D) of the GDI-DE Geodateninfrastruktur Deutschland (Spatial Data Infrastructure Germany) in joint cooperation with the 3DIM Working Group and the CityGML SWG within OGC.

For further information see http://www.sig3d.org/

The preparation of the English document version and the European discussion has been supported by the European Spatial Data Research Organization (EuroSDR; formerly known as OEEPE) in an EuroSDR Commission III project. For further information see http://www.eurosdr.net

This Document was submitted to the Open Geospatial Consortium (OGC) by the members of the CityGML 3.0 Standards Working Group of the OGC. Amongst others, this comprises the following organizations:

- Autodesk, Inc. (primary submitter)
- Bentley Systems, Inc. (primary submitter)
- Technical University Berlin (submitter of technology)
- Ordnance Survey, UK
- University of Bonn, Germany
- Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam
- Institute for Applied Computer Science, Karlsruhe Institute of Technology

CityGML was originally developed by the Special Interest Group 3D (SIG 3D), 2002 – 2012 - www.citygml.org.

v. Submitters

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

Table 1. Submission Contact Points

Name	Institution	Email
Prof. Dr. Thomas H. Kolbe, Claus Nagel, Alexandra Lorenz	Institute for Geodesy and Geoinformation Science, Technical University Berlin	thomas.kolbe@tu-berlin.de claus.nagel@tu-berlin.de alexandra.lorenz@tu-berlin.de
Dr. Gerhard Gröger, Prof. Dr. Lutz Plümer, Angela Czerwinski	Institute for Geodesy and Geoinformation, University of Bonn	Groeger@ikg.uni-bonn.de Pluemer@ikg.uni-bonn.de Czerwinski@ikg.uni-bonn.de
Haik Lorenz	Autodesk, Inc.	haik.lorenz@autodesk.com
Alain Lapierre, Stefan Apfel, Paul Scarponcini	Bentley Systems, Inc.	alain.lapierre@bentley.com stefan.apfel@bentley.com paul.scarponcini@bentley.com
Carsten Rönsdorf	Ordnance Survey, Great Britain	carsten.roensdorf@ordnancesurvey.co.uk

Name	Institution	Email
Prof. Dr. Jürgen Döllner	Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam	juergen.doellner@hpi.uni-potsdam.de
Dr. Joachim Benner, Karl-Heinz Häfele	Institute for Applied Computer Science, Karlsruhe Institute of Technology	joachim.benner@kit.edu karl-heinz.haefele@kit.edu

vi. Participants in development

Table 2. Participants in Development

Name	Institution
Ulrich Gruber, Sandra Schlüter	District Administration Recklinghausen, Cadastre Department, Germany
Frank Bildstein	Rheinmetall Defence Electronics, Germany
Rüdiger Drees	T-Systems Enterprise Services GmbH, Bonn, Germany
Andreas Kohlhaas	GIStec GmbH (formerly), Germany
Frank Thiemann	Institute for Cartography and Geoinformatics, University of Hannover
Martin Degen	Cadastre Department, City of Dortmund
Heinrich Geerling	Architekturbüro Geerling, Germany
Dr. Frank Knospe	Cadastre and Mapping Department, City of Essen,
Hardo Müller	Snowflake Software Ltd., Great Britain
Martin Rechner	rechner logistic, Germany
Jörg Haist, Daniel Holweg	Fraunhofer Institute for Computer Graphics (IGD), Darmstadt, Germany
Prof. Dr. Peter A. Henning	Faculty for Computer Science, University of Applied Sciences, Karlsruhe, Germany
Rolf Wegener, Stephan Heitmann	State Cadastre and Mapping Agency of North-Rhine Westphalia, Germany
Prof. Dr. Marc-O. Löwner	Institute for Geodesy and Photogrammetry, Technical University of Braunschweig
Dr. Egbert Casper	Zerna Ingenieure, Germany
Christian Dahmen	con terra GmbH, Germany
Nobuhiro Ishimaru, Kishiko Maruyama, Eiichiro Umino, Takahiro Hirose	Hitachi, Ltd., Japan

Name	Institution
Linda van den Brink	Geonovum, The Netherlands
Ron Lake, David Burggraf	Galdos Systems Inc., Canada
Marie-Lise Vautier, Emmanuel Devys	Institut géographique national, France
Mark Pendlington	Ordnance Survey, Great Britain

vii. Acknowledgements

The SIG 3D wishes to thank the members of the CityGML Standards Working Group and the 3D Information Management (3DIM) Working Group of the OGC as well as all contributors of change requests and comments. In particular: Tim Case, Scott Simmons, Paul Cote, Clemens Portele, Jeffrey Bell, Chris Body, Greg Buehler, François Golay, John Herring, Jury Konga, Kai-Uwe Krause, Gavin Park, Richard Pearsall, George Percivall, Mauro Salvemini, Alessandro Triglia, David Wesloh, Tim Wilson, Greg Yetman, Jim Farley, Cliff Behrens, Lukas Herman, Danny Kita, and Simon Cox.

Further credits for careful reviewing and commenting of this document go to: Ludvig Emgard, Bettina Petzold, Dave Capstick, Mark Pendlington, Alain Lapierre, and Frank Steggink.

Chapter 1. Introduction

1.1. Motivation

An increasing number of cities and companies are building virtual 3D city models for different application areas like urban planning, mobile telecommunication, disaster management, 3D cadastre, tourism, vehicle and pedes-trian navigation, facility management and environmental simulations. Furthermore, in the implementation of the European Environmental Noise Directive (END, 2002/49/EC) 3D geoinformation and 3D city models play an important role.

In recent years, most virtual 3D city models have been defined as purely graphical or geometrical models, neglecting the semantic and topological aspects. Thus, these models could almost only be used for visualisation purposes but not for thematic queries, analysis tasks, or spatial data mining. Since the limited reusability of models inhibits the broader use of 3D city models, a more general modelling approach had to be taken in order to satisfy the information needs of the various application fields.

CityGML is a common semantic information model for the representation of 3D urban objects that can be shared over different applications. The latter capability is especially important with respect to the cost-effective sustain-able maintenance of 3D city models, allowing the possibility of selling the same data to customers from different application fields. The targeted application areas explicitly include city planning, architectural design, tourist and leisure activities, environmental simulation, mobile telecommunication, disaster management, homeland securi-ty, real estate management, vehicle and pedestrian navigation, and training simulators.

CityGML is designed as an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is implemented as an application schema of the Geography Markup Language 3 (GML3), the extendible international standard for spatial data exchange and encoding issued by the Open Geospatial Consor-tium (OGC) and the ISO TC211. CityGML is based on a number of standards from the ISO 191xx family, the Open Geospatial Consortium, the W3C Consortium, the Web 3D Consortium, and OASIS.

CityGML defines the classes and relations for the most relevant topographic objects in cities and regional mod-els 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 generalisation hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. CityGML is applicable for large areas and small regions and can represent the terrain and 3D objects in different levels of detail simultaneously. Since either simple, single scale models without topology and few semantics or very complex multi-scale models with full topology and fine-grained semantical differenti-ations can be represented, CityGML enables lossless information exchange between different GI systems and users.

1.2. Historical background

CityGML has been developed since 2002 by the members of the Special Interest Group 3D (SIG 3D). Since 2010, this group is part of the initiative Spatial Data Infrastructure Germany (GDI-DE). Before

2010, the SIG 3D was affiliated to the initiative Geodata Infrastructure North Rhine-Westphalia (GDI NRW). The SIG 3D is an open group consisting of more than 70 companies, municipalities, and research institutions from Germany, Great Britain, Switzerland, and Austria working on the development and commercial exploitation of interoperable 3D city models and geovisualisation. Another result of the work from the SIG 3D is the proposition of the Web 3D Service (W3DS), a 3D portrayal service that is also being discussed in the Open Geospatial Consortium (OGC Doc. No. 05-019 and OGC Doc. No. 09-104r1).

A first successful implementation and evaluation of a subset of CityGML has been performed in the project “Pilot 3D” of the GDI NRW in 2005. Participants came from all over Germany and demonstrated city planning scenarios and tourist applications. By the beginning of 2006, a CityGML project within EuroSDR (European Spatial Data Research) started focusing on the European harmonisation of 3D city modelling. From June to December 2006, CityGML was employed and evaluated in the CAD/GIS/BIM thread of the OpenGIS Web Services Testbed #4 (OWS-4). Since 2008, CityGML (version 1.0.0) is an adopted OGC standard.

From that point in time, CityGML has disseminated worldwide. Many cities in Germany and in other countries in Europe provide their 3D city model in CityGML (Berlin, Cologne, Dresden and Munich, to mention only a few). In France, the project Bâti3D (IGN France) defines a profile of CityGML LOD2 and provides data from Paris and the city centres of Aix-en-Provence, Lille, Nantes and Marseille. CityGML also plays an important role in the pilot 3D project to obtain a 3D geoinformation standard and a 3D infrastructure for The Netherlands. Many cities in Europe like Monaco, Geneva, Zurich, Leewarden use CityGML LOD 2 or 3 to represent and exchange data, as well as cities in Denmark (LOD 2 and 3, partly LOD4). CityGML has strongly influenced the building model (version 2.0) of the INSPIRE initiative of the EU commission, which aims at the creation of an European spatial data infrastructure providing public sector data in an interoperable way. In Asia, the 3D city models of Istanbul (LOD 1 and 2), Doha, Katar (LOD3), and Yokohama (LOD2) are represented and exchanged in CityGML. Moreover, CityGML plays a crucial role for the 3D Spatial data infrastructure in Malaysia.

Today many commercial and academic tools support CityGML by providing import interfaces, export interfaces or both. An example is the 3D City Database which is a free and open source 3D geo database to store, represent, and manage virtual 3D city models on top of Oracle 10g R2 and 11g R1/R2 provided by the Technische Univer-sität Berlin. It fully supports CityGML and is shipped with a tool for the import and export of CityGML models. Furthermore, an open source Java class library and API for the processing of CityGML models (citygml4j) is provided by the Technische Universität Berlin. The conversion tool FME (Feature Manipulation Engine) from Safe Software Inc., which is part of the interoperability extension of ESRI’s ArcGIS, has read and write interfac-es for CityGML. The same applies to CAD tools as BentleyMap from Bentley Systems as well as to GIS tools like SupportGIS from CPA Geo-Information. Many 3D viewers (which all are freely available) provide read interfaces for CityGML: the Aristoteles Viewer from the University of Bonn, LandXplorer CityGML Viewer from Autodesk Inc. (the studio version for authoring and management is not free) and the FZKViewer for IFC and CityGML from KIT Karlsruhe and BS Contact from Bitmanagement Software GmbH which offers a CityGML plugin for the geospatial extension BS Contact Geo. This enumeration of software tools is not exhaus-tive and steadily growing. Please refer to the official website of CityGML at http://www.citygml.org as well as the CityGML Wiki at http://www.citygmlwiki.org for more information.

1.3. Additions in CityGML 2.0

CityGML 2.0 is a major revision of the previous version 1.0 of this International Standard (OGC Doc. No. 08-007r1), and introduces substantial additions and new features to the thematic model of CityGML. The revision was originally planned to be a minor update to version 1.1. The main endeavor of the revision process was to ensure backwards compatibility both on the level of the conceptual model and on the level of CityGML instance documents. However, some changes could not be implemented consistent with directives for minor revisions and backwards compatibility as enforced by OGC policy (cf. OGC Doc. No. 135r11). The major version number change to 2.0 is therefore a consequence of conforming to the OGC versioning policy without having to abandon any changes or additions which reflect requests from the CityGML community.

CityGML 2.0 is backwards compatible with version 1.0 in the following sense: each valid 1.0 instance is a valid 2.0 instance provided that the CityGML namespaces and schema locations in the document are changed to their actual 2.0 values. This step is required because the CityGML version number is encoded in these values, but no further actions have to be taken. Hence, there is a simple migration path from existing CityGML 1.0 instances to valid 2.0 instances.

The following clauses provide an overview of what is new in CityGML 2.0.

New thematic modules for the representation of bridges and tunnels

Bridges and tunnels are important objects in city and landscape models. They are an essential part of the trans-portation infrastructure and are often easily recognizable landmarks of a city. CityGML 1.0 has been lacking thematic modules dedicated to bridges and tunnels, and thus such objects had to be modelled and exchanged using a GenericCityObject as proxy (cf. chapter 10.12). CityGML 2.0 now introduces two new thematic modules for the explicit representation of bridges and tunnels which complement the thematic model of CityGML: the Bridge module (cf. chapter 10.4) and the Tunnel module (cf. chapter 10.5).

Bridges and tunnels can be represented in LOD 1 – 4 and the underlying data models have a coherent structure with the Building model. For example, bridges and tunnels can be decomposed into parts, thematic boundary surfaces with openings are available to semantically classify parts of the shell, and installations as well as interi-or built structures can be represented. This coherent model structure facilitates the similar understanding of semantic entities and helps to reduce software implementation efforts. Both the Bridge and the Tunnel model introduce further concepts and model elements which are specific to bridges and tunnels respectively.

Additions to existing thematic modules

- *CityGML Core module (cf. chapter 10.1)* Two new optional attributes have been added to the abstract base class *core:_CityObject* within the *CityGML Core* module: *relativeToTerrain* and *relativeToWater*. These attributes denote the feature's location with respect to the terrain and water surface in a qualitative way, and thus facilitate simple and efficient queries (e.g., for the number of subsurface buildings) without the need for an additional digital terrain model or a model of the water body.
- *Building module (cf. chapter 10.3)*
  - *LOD0 representation :* Buildings can now be represented in LOD0 by footprint and/or roof

edge polygons. This al-lows the easy integration of existing 2D data and of roof reconstructions from aerial and satel-lite imagery into a 3D city model. The representations are restricted to horizontal, 3-dimensional surfaces.

    - *Additional thematic boundary surfaces :* In order to semantically classify parts of the outer building shell which are neither horizontal wall surfaces nor parts of the roof, two additional boundary surfaces are introduced: *OuterFloorSurface* and *OuterCeilingSurface*.
    - *Additional relations to thematic boundary surfaces :* In addition to *_ AbstractBuilding* and *Room*, the surface geometries of *BuildingInstallation* and *IntBuildingInstallation* features can now be semantically classified using thematic boundary surfaces. For example, this facilitates the semantic differentiation between roof and wall sur-faces of dormers which are modeled as *BuildingInstallation*.
    - *Additional use of implicit geometries :* Implicit geometries (cf. chapter 8.3) are now available for the representation of *_ Opening*, *BuildingInstallation*, and *IntBuildingInstallation* in addition to *BuildingFurniture*. A prototyp-ical geometry for these city objects can thus be stored once and instantiated at different loca-tions in the 3D city model.

- *Generics module (cf. chapter 10.12)* Two generic attributes have been added to the *Generics* module: *MeasureAttribute* and *GenericAttributeSet*. A *MeasureAttribute* facilitates the representation of measured values together with a reference to the employed unit. A *GenericAttributeSet* is a named collection of arbitrary generic attributes. It provides an optional *codeSpace* attribute to denote the authority organization who defined the attribute set.
- *LandUse module (cf. chapter 10.10)* The scope of the feature type *LandUse* has been broadened to comprise both areas of the earth's surface dedicated to a specific land use and areas of the earth's surface having a specific land cover with or without vegetation.
- *Attributes* class, function, *and* usage *(all modules)* In order to harmonize the use of the attributes *class*, *function*, and *usage*, this attribute triplet has been complemented for all feature classes that at least provided one of the attributes in CityGML 1.0.

Additions to the CityGML code list mechanism

In CityGML, code lists providing the allowed values for enumerative attributes such as *class*, *function*, and *usage* can be specified outside the CityGML schema by any organization or information community according to their specific information needs. This mechanism is, however, not fully reflected in the CityGML 1.0 encoding schema, because in a CityGML 1.0 instance document a corresponding attribute cannot point to the dictionary with the used code list values. This has been corrected for CityGML 2.0: All attributes taking values from code lists are now of type *gml:CodeType* following the GML 3.1.1 mechanism for the encoding of code list values (cf.chapter 10.14 for more information). The *gml:CodeType* adds an optional *codeSpace* value to enumerative attributes which allows for providing a persistent URI pointing to the corresponding dictionary.

Changelog for CityGML 2.0

Changes on the level of XML schema components are provided in Annex F.

Further edits to the specification document

- *Accuracy requirements for Levels of Detail (LOD) (cf. chapter 6.2)* The accuracy requirements for the different CityGML LODs proposed in chapter 6.2 are non-normative. The wording of chapter

6.2 in CityGML 1.0 is however inconsistent with regard to this fact and thus has been clarified for CityGML 2.0.

- *Rework of the CityGML example datasets (cf. Annex G)* The CityGML examples provided in Annex G have been reworked and extended. They now show a consistent building model in all five LODs and demonstrate, for example, the semantic and geometric refinement of the building throughout the different LODs as well as the usage of XLinks to share geom-etry elements between features. The datasets are shipped with the CityGML XML Schema package, and are available at http://schemas.opengis.net/citygml/examples/2.0/.
- *New example for the usage of Application Domain Extensions (cf. Annex I)* A second example for the usage of Application Domain Extensions in the field of Ubiquitous Network Robots Services has been added in Annex I.

Chapter 2. Scope

This document is an OGC Encoding Standard for the representation, storage and exchange of virtual 3D city and landscape models. CityGML is implemented as an application schema of the Geography Markup Language version 3.1.1 (GML3).

CityGML models both complex and georeferenced 3D vector data along with the semantics associated with the data. In contrast to other 3D vector formats, CityGML is based on a rich, general purpose information model in addition to geometry and appearance information. For specific domain areas, CityGML also provides an exten-sion mechanism to enrich the data with identifiable features under preservation of semantic interoperability.

Targeted application areas explicitly include urban and landscape planning; architectural design; tourist and leisure activities; 3D cadastres; environmental simulations; mobile telecommunications; disaster management; homeland security; vehicle and pedestrian navigation; training simulators and mobile robotics.

CityGML is considered a source format for 3D portraying. The semantic information contained in the model can be used in the styling process which generates computer graphics represented e.g. as KML/COLLADA or X3D files. The appropriate OGC Portrayal Web Service for this process is the OGC Web 3D Service (W3DS). An image-based 3D portrayal service for virtual 3D landscape and city models is provided by the OGC Web View Service (WVS).

Features of CityGML:

- Geospatial information model (ontology) for urban landscapes based on the ISO 191xx family
- GML3 representation of 3D geometries, based on the ISO 19107 model
- Representation of object surface characteristics (e.g. textures, materials)
- Taxonomies and aggregations
    - Digital Terrain Models as a combination of (including nested) triangulated irregular networks (TINs), regular rasters, break and skeleton lines, mass points
    - Sites (currently buildings, bridges, and tunnels)
    - Vegetation (areas, volumes, and solitary objects with vegetation classification)
    - Water bodies (volumes, surfaces)
    - Transportation facilities (both graph structures and 3D surface data)
    - Land use (representation of areas of the earth’s surface dedicated to a specific land use)
    - City furniture
    - Generic city objects and attributes
    - User-definable (recursive) grouping
- Multiscale model with 5 well-defined consecutive Levels of Detail (LOD):
    - LOD0 – regional, landscape
    - LOD1 – city, region

- LOD2 – city districts, projects
- LOD3 – architectural models (outside), landmarks
- LOD4 – architectural models (interior)

- Multiple representations in different LODs simultaneously; generalisation relations between objects in different LODs
- Optional topological connections between feature (sub)geometries
- Application Domain Extensions (ADE): Specific “hooks” in the CityGML schema allow to define ap-plication specific extensions, for example for noise pollution simulation, or to augment CityGML by properties of the new National Building Information Model Standard (NBIMS) in the U.S.

Chapter 3. Conformance

This Best Practice defines XXXX.

Requirements for N target types are considered: * AAAA * BBBB

Conformance with this Best Practice shall be checked using all the relevant tests specified in Annex A (normative) of this document.

In order to conform to this OGC® Best Practice, a software implementation shall choose to implement: * Any one of the conformance levels specified in Annex A (normative). * Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

All requirements-classes and conformance-classes described in this document are owned by the document(s) identified.

NOTE the following is the 2.0 content

Conformance targets addressed by this International standard are CityGML instance documents only. Future revisions of this International Standard may also address consumers or producers as conformance targets. Clauses 8 to 10 of this International standard specify separate CityGML XML Schema definitions and normative aspects, i.e. CityGML modules, which shall be used in CityGML instance documents in accordance with clause 7. Implementations are not required to support the full range of capabilities provided by the universe of all CityGML modules. Valid partial implementations are supported following the rules and guidelines for CityGML profiles in chapter 7.2. CityGML instance documents claiming conformance to this International Standard shall: a) conform to the rules and requirements specified in clauses 7 to 10; b) pass all relevant test cases of the abstract test suite in annex B.1; c) satisfy all relevant conformance classes of the abstract test suite related to CityGML modules in annex B.2.

Chapter 4. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of OGC 12-019. 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 12-019 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.

The following documents are indispensable for the application of the CityGML standard. The geometry model of GML 3.1.1 is used except for some added concepts like implicit geometries (cf. chapter 8.2). The appearance model (cf. chapter 9) draws concepts from both *X3D* and *COLLADA*. Addresses are represented using the OASIS extensible Address Language *xAL*.

- ISO / TC154: ISO 8601:2004, Data elements and interchange formats – Information interchange – Representation of dates and times
- ISO / TC211: ISO/TS 19103:2005, Geographic Information – Conceptual Schema Language
- ISO / TC211: ISO 19105:2000, Geographic information – Conformance and testing
- ISO / TC211: ISO 19107:2003, Geographic Information – Spatial Schema
- ISO / TC211: ISO 19109:2005, Geographic Information – Rules for Application Schemas
- ISO / TC211: ISO 19111:2003, Geographic information – Spatial referencing by coordinates
- ISO / TC211: ISO 19115:2003, Geographic Information – Metadata
- ISO / TC211: ISO 19123:2005, Geographic Information – Coverages
- ISO / TC211: ISO/TS 19139:2007, Geographic Information – Metadata – XML schema implementation
- ISO / TC211: ISO/IEC 19775:2004, X3D Abstract Specification
- OGC: OpenGIS® Abstract Specification Topic 0, Overview, OGC document 04-084
- OGC: OpenGIS® Abstract Specification Topic 5, The OpenGIS Feature, OGC document 99-105r2
- OGC: OpenGIS® Abstract Specification Topic 8, Relations between Features, OGC document 99-108r2
- OGC: OpenGIS® Abstract Specification Topic 10, Feature Collections, OGC document 99-110
- OGC: OpenGIS® Geography Markup Language Implementation Specification, Version 3.1.1, OGC document 03-105r1
- OGC: OpenGIS® GML 3.1.1 Simple Dictionary Profile, Version 1.0.0, OGC document 05-099r2
- IETF: IETF RFC 2045 & 2046, Multipurpose Internet Mail Extensions (MIME). (November 1996)
- IETF: IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax. (August 1998)
- W3C: W3C XLink, XML Linking Language (XLink) Version 1.0. W3C Recommendation (27 June 2001)
- W3C: W3C XMLName, Namespaces in XML. W3C Recommendation (14 January 1999)
- W3C: W3C XMLSchema-1, XML Schema Part 1: Structures. W3C Recommendation (2 May 2001)

- W3C: W3C XMLSchema-2, XML Schema Part 2: Datatypes. W3C Recommendation (2 May 2001)
- W3C: W3C XPointer, XML Pointer Language (XPointer) Version 1.0. W3C Working Draft (16 August 2002)
- W3C: W3C XML Base, XML Base, W3C Recommendation (27 June 2001)
- W3C: W3C XML, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation (6 October 2000)
- OASIS (Organization for the Advancement of Structured Information Standards): extensible Address Language (xAL v2.0).
- Khronos Group Inc.: COLLADA – Digital Asset Schema Release 1.4.1
- Jelliffe, R: The Schematron Assertion Language 1.5. (2002-10-01)

Chapter 5. 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 Best Practice.

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

5.1. term name

text of the definition

5.2. Abbreviated Terms

The following abbreviated terms are used in this document:

- 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
- 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

- ISO International Organization for Standardisation
- LOD Level of Detail
- 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
- TC211 ISO Technical Committee 211
- 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

Chapter 6. Conventions

6.1. Identifiers

The normative provisions in this document are denoted by the URI

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

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

6.2. UML Notation

The CityGML standard is presented in this document in diagrams using the Unified Modeling Language (UML) static structure diagram (see Booch et al. 1997). The UML notations used in this standard are described in the diagram below Figure 1.

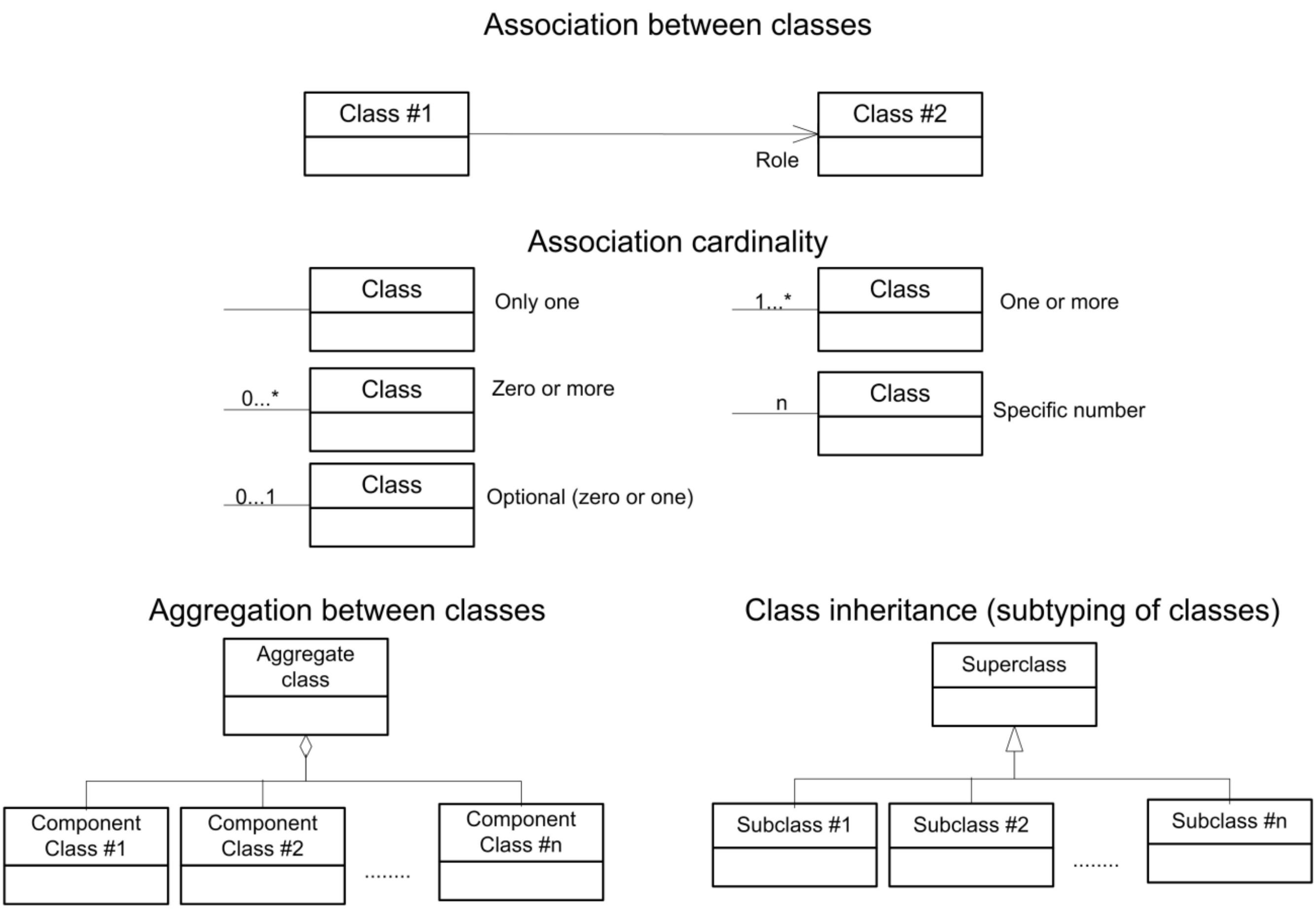


Figure 1. UML notation (see ISO TS 19103, Geographic information - Conceptual schema language).

According to GML3 all associations between model elements in CityGML are uni-directional. Thus, associations in CityGML 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:

- <<Geometry>> represents the geometry of an object. The geometry is an identifiable and distinguishable object that is derived from the abstract GML type *AbstractGeometryType*.
- <<Feature>> represents a thematic feature according to the definition in ISO 19109. A feature is an identifiable and distinguishable object that is derived from the abstract GML type *AbstractFeatureType*.
- <<Object>> represents an identifiable and distinguishable object that is derived from the abstract GML type *AbstractGMLType*.
- <<Enumeration>> enumerates the valid attribute values in a fixed list of named literal values. Enumerations are specified inline the CityGML schema.
- <<CodeList>> enumerates the valid attribute values. In contrast to Enumeration, the list of values is open and, thus, not given inline the CityGML schema. The allowed values can be provided within an external code list. It is recommended that code lists are implemented as simple dictionaries following the GML 3.1.1 Simple Dictionary Profile (cf. chapter 6.6 and chapter 10.14).
- <<Union>> is a list of attributes. The semantics are that only one of the attributes can be present at any time.
- <<PrimitiveType>> is used for representations supported by a primitive type in the implementation.
- <<DataType>> is used as a descriptor of a set of values that lack identity. Data types include primitive prede-fined types and user-definable types. A DataType is thus a class with few or no operations whose primary purpose is to hold the abstract state of another class for transmittal, storage, encoding or persistent storage.
- <<Leaf>> is used within UML package diagrams to indicate model elements that can have no further subtypes.
- <<XSDSchema>> is used within UML package diagrams to denote the root element of an XSD Schema contain-ing all the definitions for a particular namespace. All the package contents or component classes are placed within the one schema.
- <<ApplicationSchema>> is used within UML package diagrams to denote an XML Schema definition funda-mentally dependent on the concepts of another independent Standard within the XML Schema metalan-guage. For example, ApplicationSchema indicates extensions of GML consistent with the GML “rules for application schemas”.

In order to enhance the readability of the CityGML UML diagrams, classes are depicted in different colors if they belong to different UML packages (see Fig. 8 for an overview of UML packages). The following coloring scheme is applied:

- Classes painted in yellow belong to the UML package which is subject of discussion in that clause of the specification in which the UML diagram is given. For example, in the context of chapter 10.1 which introduces the *CityGML Core* module, the yellow color is used to denote classes which are de-fined in the *CityGML Core* UML package. Likewise, the yellow classes shown in UML diagrams in chapter 10.3 are associated with the *Building* module which is subject of discussion in that chapter.

- Classes painted in blue belong to a CityGML UML package different to that associated with the yellow color. In order to explicitly denote the UML package of such classes, their class names carry a namespace prefix which is uniquely associated with a CityGML module throughout this specification (cf. section 4.3 for a list of namespaces and prefixes). For example, in the context of the *Building* module, classes from the *CityGML Core* module are painted in blue and their class names are preceded by the prefix core.
- Classes painted in green are defined in GML3 and their class names are preceded by the prefix *gml*.

The following example UML diagram demonstrates the UML notation and coloring scheme used throughout this specification. In this example, the yellow classes are associated with the *CityGML Building* module, the blue classes are from the *CityGML Core* module, and the green class depicts a geometry element defined by GML3.

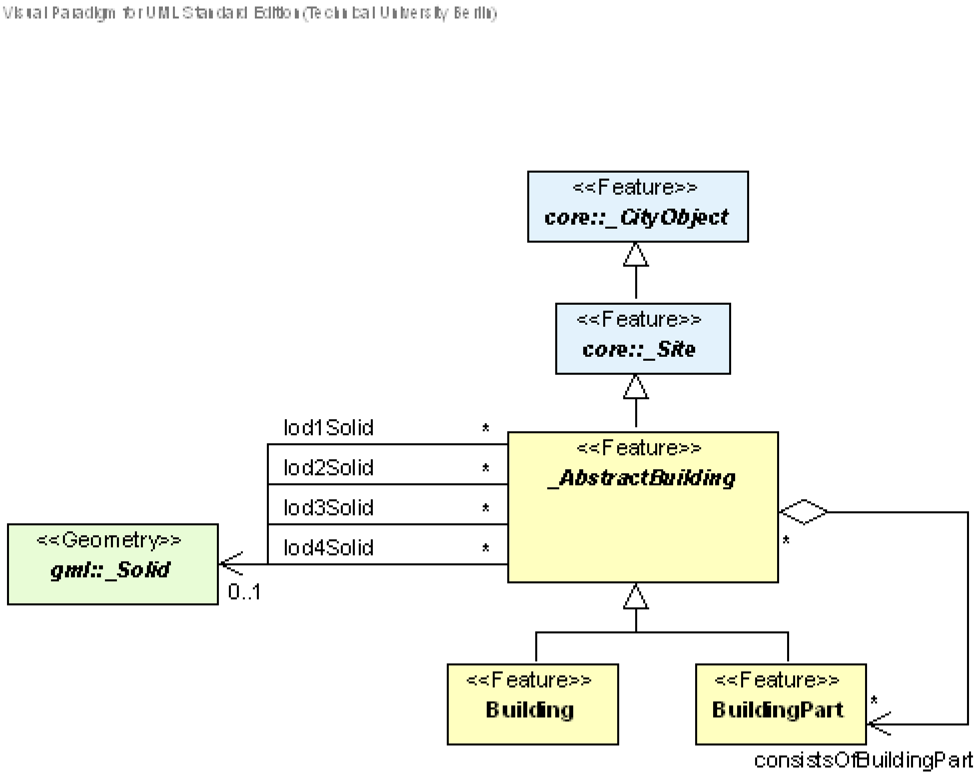


Figure 2. Example UML diagram demonstrating the UML notation and coloring scheme used throughout the CityGML specification.

6.3. XML namespaces and namespace prefixes

The CityGML data model is thematically decomposed into a core module and thematic extension modules. All modules including the core are specified by their own XML schema file, each defining a globally unique XML namespace. The extension modules are based on the core module and, thus, contain (by reference) the CityGML core schema.

Within this document the module namespaces are associated with recommended prefixes. These prefixes are consistently used within the normative parts of this specification, for all UML diagrams and example CityGML instance documents. The CityGML core and extension modules along with their XML namespace identifiers and recommended namespace prefixes are listed in Tab. 1.

Table 3. List of CityGML modules, their associated XML namespace identifiers, and example namespace prefixes.

CityGML module	Namespace identifier	Namespace prefix
CityGML Core	http://www.opengis.net/citygml/2.0	core
Appearance	http://www.opengis.net/citygml/appearance/2.0	app
Bridge	http://www.opengis.net/citygml/bridge/2.0	brid
Building	http://www.opengis.net/citygml/building/2.0	bldg
CityFurniture	http://www.opengis.net/citygml/cityfurniture/2.0	frn
CityObjectGroup	http://www.opengis.net/citygml/cityobjectgroup/2.0	grp
Generics	http://www.opengis.net/citygml/generics/2.0	gen
LandUse	http://www.opengis.net/citygml/landuse/2.0	luse
Relief	http://www.opengis.net/citygml/relief/2.0	dem
Transportation	http://www.opengis.net/citygml/transportation/2.0	tran
Tunnel	http://www.opengis.net/citygml/tunnel/2.0	tun
Vegetation	http://www.opengis.net/citygml/vegetation/2.0	veg
WaterBody	http://www.opengis.net/citygml/waterbody/2.0	wtr
TexturedSurface [deprecated]	http://www.opengis.net/citygml/texturedsurface/2.0	tex

Further XML Schema definitions relevant to this standard are shown in Tab. 2 along with the corresponding XML namespace identifiers and namespace prefixes consistently used within this document.

Table 4. List of XML Schema definitions, their associated XML namespace identifiers, and example namespace prefixes used within this document.

XML Schema Definition	Namespace identifier	Namespace prefix
Geography Markup Language version 3.1.1 (from OGC)	http://www.opengis.net/gml	gml

XML Schema Definition	Namespace identifier	Namespace prefix
Extensible Address Language version 2.0 (from OASIS)	urn:oasis:names:tc:ciq:xsdschema:xAL:2.0	xAL
Schematron Assertion Lan-guage version 1.5	http://www.ascc.net/xml/schematron	sch

Chapter 7. Overview of CityGML

CityGML is an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is an application schema for the Geography Markup Language version 3.1.1 (GML3), the extendible interna-tional standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211.

The aim of the development of CityGML is to reach a common definition of the basic entities, attributes, and relations of a 3D city model. This is especially important with respect to the cost-effective sustainable mainte-nance of 3D city models, allowing the reuse of the same data in different application fields.

CityGML not only represents the graphical appearance of city models but specifically addresses the representa-tion of the semantic and thematic properties, taxonomies and aggregations. CityGML includes a geometry model and a thematic model. The geometry model allows for the consistent and homogeneous definition of geometrical and topological properties of spatial objects within 3D city models (chapter 8). The base class of all objects is _ *CityObject* which is a subclass of the GML class _ *Feature*. All objects inherit the properties from _ *CityObject*.

The thematic model of CityGML employs the geometry model for different thematic fields like Digital Terrain Models, sites (i.e. buildings, bridges, and tunnels), vegetation (solitary objects and also areal and volumetric biotopes), land use, water bodies, transportation facilities, and city furniture (chapter 10). Further objects, which are not explicitly modelled yet, can be represented using the concept of generic objects and attributes (chapter 6.11). In addition, extensions to the CityGML data model applying to specific application fields can be realised using the Application Domain Extensions (ADE) (chapter 6.12). Spatial objects of equal shape which appear many times at different positions like e.g. trees, can also be modelled as prototypes and used multiple times in the city model (chapter 8.2). A grouping concept allows the combination of single 3D objects, e.g. buildings to a building complex (chapter 6.8). Objects which are not geometrically modelled by closed solids can be virtually sealed in order to compute their volume (e.g. pedestrian underpasses, tunnels, or airplane hangars). They can be closed using *ClosureSurfaces* (chapter 6.4). The concept of the *TerrainIntersectionCurve* is introduced to inte-grate 3D objects with the Digital Terrain Model at their correct positions in order to prevent e.g. buildings from floating over or sinking into the terrain (chapter 6.5).

CityGML differentiates five consecutive Levels of Detail (LOD), where objects become more detailed with increasing LOD regarding both their geometry and thematic differentiation (chapter 6.2). CityGML files can - but do not have to - contain multiple representations (and geometries) for each object in different LOD simulta-neously. Generalisation relations allow the explicit representation of aggregated objects over different scales.

In addition to spatial properties, CityGML features can be assigned appearances. Appearances are not limited to visual data but represent arbitrary observable properties of the feature’s surface such as infrared radiation, noise pollution, or earthquake-induced structural stress (chapter 9).

Furthermore, objects can have external references to corresponding objects in external datasets (chapter 6.7). The possible attribute values of enumerative object attributes can be enumerated in code lists defined in external, redefinable dictionaries (chapter 6.6).

Chapter 8. General characteristics of CityGML

8.1. Modularisation

The CityGML data model consists of class definitions for the most important types of objects within virtual 3D city models. These classes have been identified to be either required or important in many different application areas. However, implementations are not required to support the overall CityGML data model in order to be conformant to the standard, but may employ a subset of constructs according to their specific information needs. For this purpose, modularisation is applied to the CityGML data model (cf. chapter 7).

The CityGML data model is thematically decomposed into a *core module* and thematic *extension modules*. The core module comprises the basic concepts and components of the CityGML data model and, thus, must be implemented by any conformant system. Based on the core module, each extension covers a specific thematic field of virtual 3D city models. CityGML introduces the following thirteen thematic extension modules: *Appearance*, *Bridge*, *Building*, *CityFurniture*, *CityObjectGroup*, *Generics*, *LandUse*, *Relief*, *Transportation*, *Tunnel*, *Vegetation*, *WaterBody*, and *TexturedSurface* [deprecated].

CityGML compliant implementations may support any combination of extension modules in conjunction with the core module. Such combinations of modules are called CityGML profiles. Therefore, CityGML profiles allow for valid partial implementations of the overall CityGML data model.

8.2. Multi-scale modelling (5 levels of detail, LOD)

CityGML supports different Levels of Detail (LOD). LODs are required to reflect independent data collection processes with differing application requirements. Further, LODs facilitate efficient visualisation and data analysis (see Fig. 3). In a CityGML dataset, the same object may be represented in different LOD simultaneously, enabling the analysis and visualisation of the same object with regard to different degrees of resolution. Furthermore, two CityGML data sets containing the same object in different LOD may be combined and integrated. However, it will be within the responsibility of the user or application to make sure objects in different LODs refer to the same real-world object.

The coarsest level LOD0 is essentially a two and a half dimensional Digital Terrain Model over which an aerial image or a map may be draped. Buildings may be represented in LOD0 by footprint or roof edge polygons. LOD1 is the well-known blocks model comprising prismatic buildings with flat roof structures. In contrast, a building in LOD2 has differentiated roof structures and thematically differentiated boundary surfaces. LOD3 denotes architectural models with detailed wall and roof structures potentially including doors and windows. LOD4 completes a LOD3 model by adding interior structures for buildings. For example, buildings in LOD4 are composed of rooms, interior doors, stairs, and furniture. In all LODs appearance information such as highresolution textures can be mapped onto the structures (cf. 6.9).

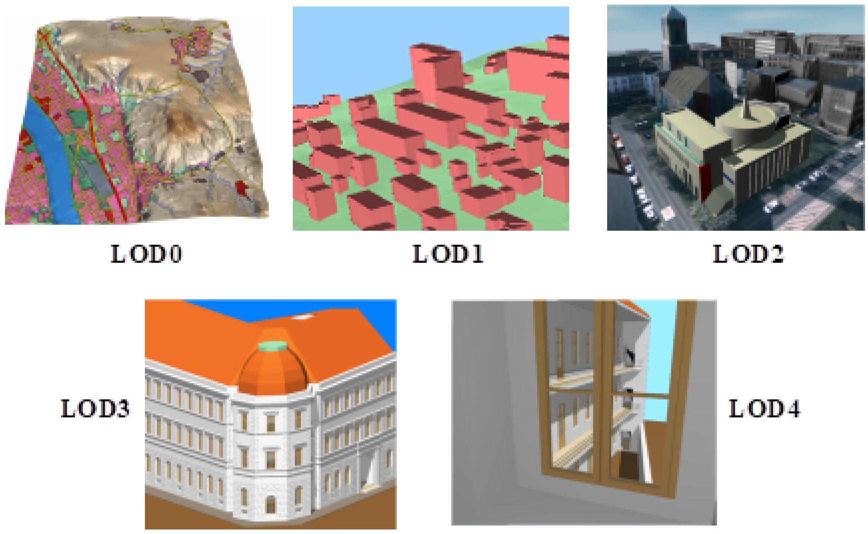


Figure 3. The five levels of detail (LOD) defined by CityGML (source: IGG Uni Bonn)

LODs are also characterised by differing accuracies and minimal dimensions of objects (cf. Tab. 3). The accura-cy requirements given in this standard are debatable and are to be considered as discussion proposals. Accuracy is described as standard deviation of the absolute 3D point coordinates. Relative 3D point accuracy will be added in a future version of CityGML and it is typically much higher than the absolute accuracy. In LOD1, the positional and height accuracy of points should be 5m or less, while all objects with a footprint of at least 6m by 6m should be considered. The positional and height accuracy of LOD2 is proposed to be 2m or better. In this LOD, all objects with a footprint of at least 4m × 4m should be considered. Both types of accuracies in LOD3 should be 0.5m, and the minimal footprint is suggested to be 2m × 2m. Finally, the positional and height accura-cy of LOD4 should be 0.2m or less. By means of these figures, the classification in five LOD may be used to assess the quality of 3D city model datasets. The LOD categorisation makes datasets comparable and provides support for their integration.

Table 5. LOD 0-4 of CityGML with their proposed accuracy requirements (discussion proposal, based on: Albert et al. 2003).

	LOD0	**LOD1**	**LOD2**	**LOD3**	**LOD4**
Model scale description	regional, landscape	city, region	city, city districts, projects	city districts, architectural models (exterior), landmark	architectural models (interi-or), landmark
Class of accuracy	lowest	low	middle	high	very high
Absolute 3D point accuracy (position / height)	lower than LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m

	LOD0	LOD1	LOD2	LOD3	LOD4
Generalisation	maximal generalisation	object blocks as generalised features; > 6*6m/3m	objects as generalised features; > 4*4m/2m	object as real features; > 2*2m/1m	constructive elements and openings are represented
Building installations	no	no	yes	representative exterior features	real object form
Roof structure/representation	yes	flat	differentiated roof structures	real object form	real object form
Roof overhanging parts	yes	no	yes, if known	yes	yes
CityFurniture	no	important objects	prototypes, generalized objects	real object form	real object form
SolitaryVegetationObject	no	important objects	prototypes, higher 6m	prototypes, higher 2m	prototypes, real object form
PlantCover	no	>50*50m	>5*5m	< LOD2	<LOD2
…to be continued for the other feature themes					

Whereas in CityGML each object can have a different representation for every LOD, often different objects from the same LOD will be generalised to be represented by an aggregate object in a lower LOD. CityGML supports the aggregation / decomposition by providing an explicit generalisation association between city objects (further details see UML diagram in chapter 10.1).

8.3. Coherent semantical-geometrical modelling

One of the most important design principles for CityGML is the coherent modelling of semantics and geomet-rical/topological properties. At the semantic level, real-world entities are represented by features, such as build-ings, walls, windows, or rooms. The description also includes attributes, relations and aggregation hierarchies (part-whole-relations) between features. Thus the part-of-relationship between features can be derived at the semantic level only, without considering geometry. However, at the spatial level, geometry objects are assigned to features representing their spatial location and extent. So the model consists of two hierarchies: the semantic and the geometrical in which the corresponding objects are linked by relationships (cf. Stadler & Kolbe 2007). The advantage of this approach is that it can be navigated in both hierarchies and between both hierarchies arbitrarily, for answering thematic and/or geometrical queries or performing analyses.

If both hierarchies exist for a specific object, they must be coherent (i.e. it must be ensured that they match and fit together). For example, if a wall of a building has two windows and a door on the semantic level, then the geometry representing the wall must contain also the geometry parts of both windows and the door.

8.4. Closure surfaces

Objects, which are not modelled by a volumetric geometry, must be virtually closed in order to compute their volume (e.g. pedestrian underpasses or airplane hangars). They can be sealed using a ClosureSurface. These are special surfaces, which are taken into account, when needed to compute volumes and are neglected, when they are irrelevant or not appropriate, for example in visualisations.

The concept of ClosureSurface is also employed to model the entrances of subsurface objects. Those objects like tunnels or pedestrian underpasses have to be modelled as closed solids in order to compute their volume, for example in flood simulations. The entrances to subsurface objects also have to be sealed to avoid holes in the digital terrain model Figure 4. However, in close-range visualisations the entrance must be treated as open. Thus, closure surfaces are an adequate way to model those entrances.

NOTE Combine Figures 4

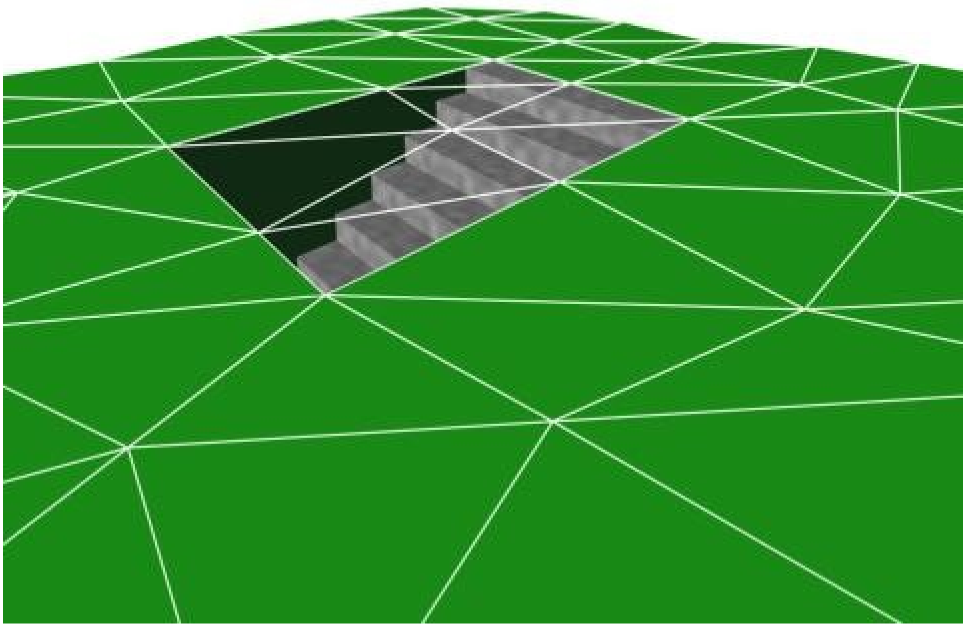


Figure 4. Closure surfaces to seal open structures. Passages are subsurface objects (left). The entrance is sealed by a virtual ClosureSurface, which is both part of the DTM and the subsurface object (right) (graphic: IGG Uni Bonn).

8.5. Terrain Intersection Curve (TIC)

A crucial issue in city modelling is the integration of 3D objects and the terrain. Problems arise if 3D objects float over or sink into the terrain. This is particularly the case if terrains and 3D objects in different LOD are combined, or if they come from different providers (Kolbe and Gröger 2003). To overcome this problem, the TerrainIntersectionCurve (TIC) of a 3D object is introduced. These curves denote the exact position, where the terrain touches the 3D object (see Fig. 5). TICs can be applied to buildings and building parts (cf. chapter 10.3), bridge, bridge parts and bridge construction elements (cf. chapter 10.5), tunnel and tunnel parts (cf. chapter 10.4), city furniture objects (cf. chapter 10.9), and generic city objects (cf. chapter 10.12). If, for example, a building has a courtyard, the TIC consists of two closed rings: one ring representing the courtyard boundary, and one which describes the building's outer boundary. This information can be used to integrate the building and a terrain by 'pulling up' or 'pulling down' the surrounding terrain to fit the TerrainIntersectionCurve. The DTM may be locally warped to fit the TIC. By this means, the TIC also ensures the correct positioning of textures or the matching of object textures with the DTM. Since the intersection with the terrain may differ depending on the LOD, a 3D object may have different TerrainIntersectionCurves for all LOD.

NOTE	Combine figures 5

Figure 5. TerrainIntersectionCurve for a building (left, black) and a tunnel object (right, white). The tunnel's hollow space is sealed by a triangulated ClosureSurface (graphic: IGG Uni Bonn).

8.6. Code lists for enumerative attributes

CityGML feature types often include attributes whose values can be enumerated in a list of discrete values. An example is the attribute roof type of a building, whose attribute values typically are saddle back roof, hip roof, semi-hip roof, flat roof, pent roof, or tent roof. If such an attribute is typed as string, misspellings or different names for the same notion obstruct interoperability. Moreover, the list of possible attribute values often is not fixed and may substantially vary for different countries (e.g., due to national law and regulations) and for differ-ent information communities.

In CityGML, such enumerative attributes are of type gml:CodeType and their allowed attribute values can be provided in a code list which is specified outside the CityGML schema. A code list contains coded attribute values and ensures that the same code is used for the same notion or concept. If a code list is provided for an enumerative attribute, the attribute may only take values from this list. This allows applications to validate the attribute value and thus facilitates semantic and syntactic interoperability. It is recommended that code lists are implemented as simple dictionaries following the GML 3.1.1 Simple Dictionary Profile (cf. Whiteside 2005).

The governance of code lists is decoupled from the governance of the CityGML schema and specification. Thus, code lists may be specified by any organisation or information community according to their information needs. There shall be one authority per code list who is in charge of the code list values and the maintenance of the code list. Further information on the CityGML code list mechanism is provided in chapter 10.14.

Code lists can have references to existing models. For example, room codes defined by the Open Standards Consortium for Real Estate (OSCRE) can be referenced or classifications of buildings and building parts intro-duced by the National Building Information Model Standard (NBIMS) can be used. Annex C contains non-normative code lists proposed by the SIG 3D for almost all enumerative attributes in CityGML. They can be directly referenced in CityGML instance documents and serve as an example for the definition of code lists.

8.7. External references

3D objects are often derived from or have relations to objects in other databases or data sets. For example, a 3D building model may have been constructed from a two-dimensional footprint in a cadastre data set, or may be derived from an architectural model (Fig. 6). The reference of a 3D object to its corresponding object in an external data set is essential, if an update must be propagated or if additional data is required, for example the name and address of a building's owner in a cadastral information system or information on antennas and doors in a facility management system. In order to supply such information, each _CityObject may refer to external data sets (for the UML diagram see Fig. 21; and for XML schema definition see annex A.1) using the concept of ExternalReference. Such a reference denotes the external information system and the unique identifier of the object in this system. Both are specified as a Uniform Resource Identifier (URI), which is a generic format for references to any kind of resources on the internet. The generic concept of external references allows for any _CityObject an arbitrary number of links to corresponding objects in external information systems (e.g. ALKIS, ATKIS, OS MasterMap®, GDF, etc.).

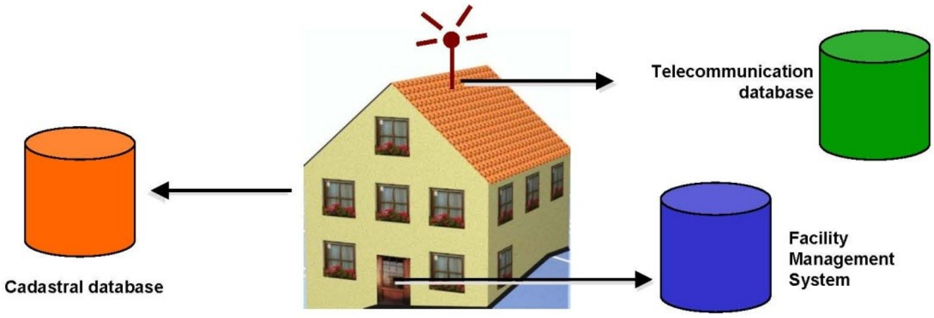


Figure 6. External references (graphic: IGG Uni Bonn).

8.8. City object groups

The grouping concept of CityGML allows for the aggregation of arbitrary city objects according to user-defined criteria, and to represent and transfer these aggregations as part of a city model (for the UML diagram see chapter 10.11; XML schema definition see annex A.6). A group may be assigned one or more names and may be further classified by specific attributes, for example, "escape route from room no. 43 in house no. 1212 in a fire scenario" as a name and "escape route" as type. Each member of the group can optionally be assigned a role name, which specifies the role this particular member plays in the group. This role name may, for example, describe the sequence number of this object in an escape route, or in the case of a building complex, denote the main building.

A group may contain other groups as members, allowing nested grouping of arbitrary depth. The grouping concept is delivered by the thematic extension module CityObjectGroup of CityGML (cf. chapter 10.11).

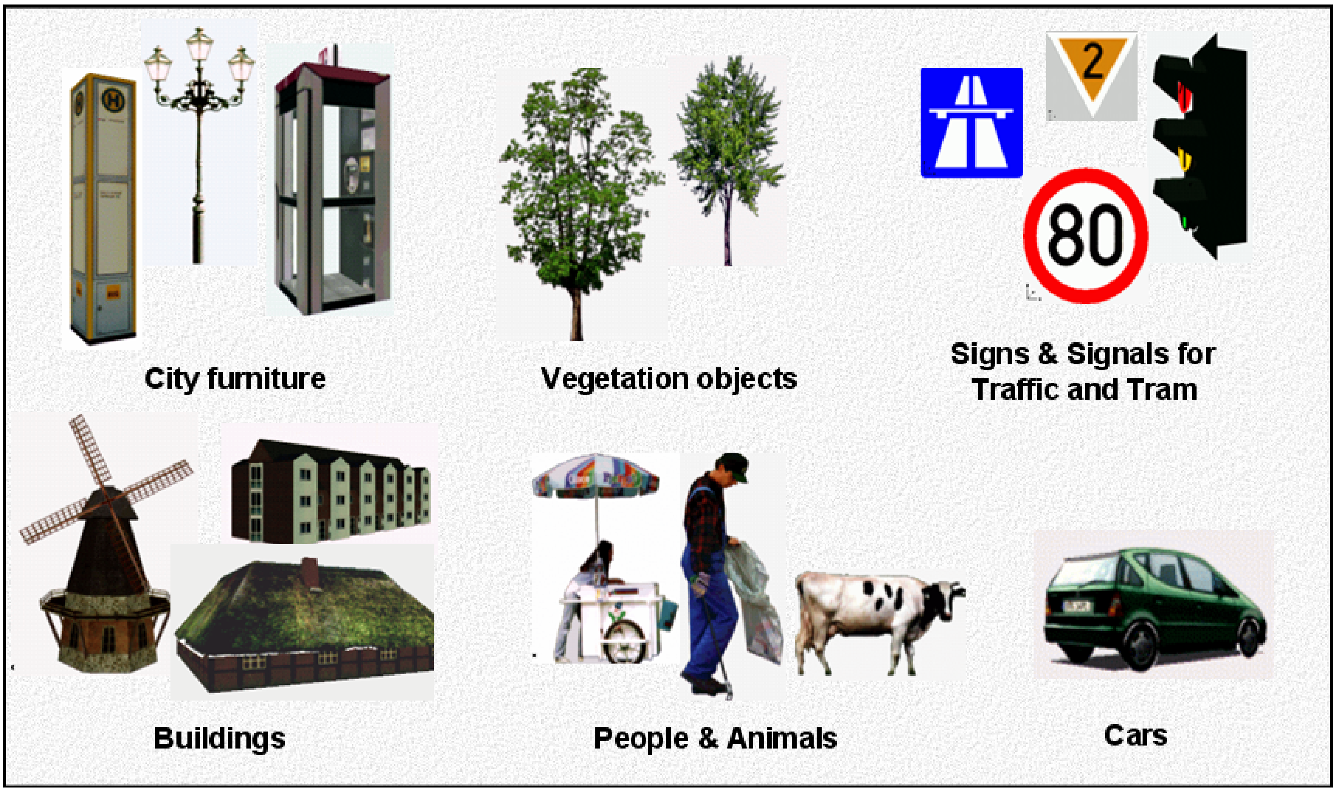
8.9. Appearances

Information about a surface’s appearance, i.e. observable properties of the surface, is considered an integral part of virtual 3D city models in addition to semantics and geometry. Appearance relates to any surface-based theme, e.g. infrared radiation or noise pollution, not just visual properties. Consequently, data provided by appearances can be used as input for both presentation of and analysis in virtual 3D city models.

CityGML supports feature appearances for an arbitrary number of themes per city model. Each LOD of a feature can have an individual appearance. Appearances can represent – among others – textures and georeferenced textures. CityGML’s appearance model is packaged within its own extension module Appearance (cf. chapter 9).

8.10. Prototypic objects / scene graph concepts

In CityGML, objects of equal shape like trees and other vegetation objects, traffic lights and traffic signs can be represented as prototypes which are instantiated multiple times at different locations (Fig. 7). The geometry of prototypes is defined in local coordinate systems. Every instance is represented by a reference to the prototype, a base point in the world coordinate reference system and a transformation matrix that facilitates scaling, rotation, and translation of the prototype. The principle is adopted from the concept of scene graphs used in computer graphics standards like VRML and X3D. As the GML3 geometry model does not provide support for scene graph concepts, it is implemented as an extension to the GML3 geometry model (for further description cf. chapter 8.2).



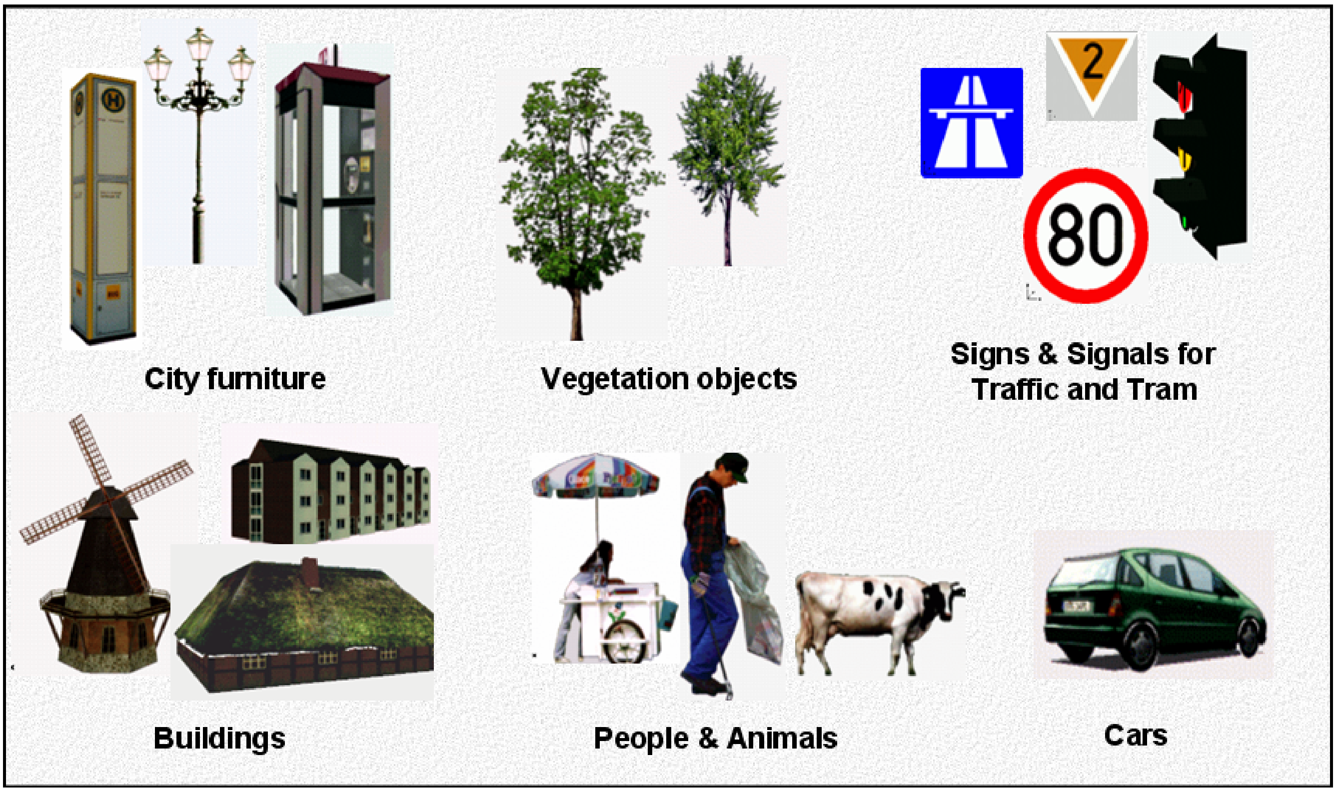


Figure 7. Examples of prototypic shapes (source: Rheinmetall Defence Electronics).

8.11. Generic city objects and attributes

CityGML is being designed as a universal topographic information model that defines object types and attributes which are useful for a broad range of applications. In practical applications the objects within specific 3D city models will most likely contain attributes which are not explicitly modelled in CityGML. Moreover, there might be 3D objects which are not covered by the thematic classes of CityGML. CityGML provides two different concepts to support the exchange of such data: 1) generic objects and attributes, and 2) Application Domain Extensions (cf. chapter 6.12).

The concept of generic objects and attributes allows for the extension of CityGML applications during runtime, i.e. any _CityObject may be augmented by additional attributes, whose names, data types, and values can be provided by a running application without any change of the CityGML XML schema. Similarly, features not represented by the predefined thematic classes of the CityGML data model may be modelled and exchanged using generic objects. The generic extensions of CityGML are provided by the thematic extension module Generics (cf. chapter 10.12).

The current version of CityGML does not include, for example, explicit thematic models for embankments, excavations and city walls. These objects may be stored or exchanged using generic objects and attributes.

8.12. Application Domain Extensions (ADE)

Application Domain Extensions (ADE) specify additions to the CityGML data model. Such additions comprise the introduction of new properties to existing CityGML classes like e.g. the number of habitants of a building or the definition of new object types. The difference between ADEs and generic objects and attributes is, that an ADE has to be defined in an extra XML schema definition file with its own namespace. This file has to explicitly import the XML Schema definition of the extended CityGML modules.

The advantage of this approach is that the extension is formally specified. Extended CityGML instance docu-ments can be validated against the CityGML and the respective ADE schema. ADEs can be defined (and even standardised) by information communities which are interested in specific application fields. More than one ADE can be actively used in the same dataset (further description cf. chapter 10.13).

ADEs may be defined for one or even several CityGML modules providing a high flexibility in adding additional information to the CityGML data model. Thus, the ADE mechanism is orthogonally aligned with the modular-isation approach of CityGML. Consequently, there is no separate extension module for ADEs.

In this specification, two examples for ADEs are included:

- An ADE for Noise Immission Simulation (Annex H) which is employed in the simulation of environmental noise dispersion according to the Environmental Noise Directive of the European Commission (2002/49/EC);
- An ADE for Ubiquitous Network Robots Services (Annex I) which demonstrates the usage of CityGML for the navigation of robots in indoor environments.

Further examples for ADEs are the CAFM ADE (Bleifuß et al., 2009) for facility management, the UtilityNetworkADE (Becker et al., 2011) for the integrated 3D modeling of multi-utility networks and their interdependencies, the HydroADE (Schulte and Coors, 2008) for hydrographical applications and the GeoBIM (IFC) ADE (van Berlo et al., 2011) which combines BIM information from IFC (from bSI) with CityGML and is implemented in the open source modelserver BIMserver.org.

Chapter 9. Modularization

CityGML is a rich standard both on the thematic and geometric-topological level of its data model. On its the-matic level CityGML defines classes and relations for the most relevant topographic objects in cities and region-al models comprising built structures, elevation, vegetation, water bodies, city furniture, and more. In addition to geometry and appearance content these thematic components allow to employ virtual 3D city models for sophis-ticated analysis tasks in different application domains like simulations, urban data mining, facility management, and thematic inquiries.

CityGML is to be seen as a framework giving geospatial 3D data enough space to grow in geometrical, topologi-cal and semantic aspects over its lifetime. Thus, geometry and semantics of city objects may be flexibly struc-tured covering purely geometric datasets up to complex geometric-topologically sound and spatio-semantically coherent data. By this means, CityGML defines a single object model and data exchange format applicable to consecutive process steps of 3D city modelling from geometry acquisition, data qualification and refinement to preparation of data for specific end-user applications, allowing for iterative data enrichment and lossless information exchange (cf. Kolbe et al. 2009).

According to this idea of a framework, applications are not required to support all thematic fields of CityGML in order to be compliant to the standard, but may employ a subset of constructs corresponding to specific relevant requirements of an application domain or process step. The use of logical subsets of CityGML limits the com-plexity of the overall data model and explicitly allows for valid partial implementations. As for version 2.0 of the CityGML standard, possible subsets of the data model are defined and embraced by so called CityGML mod-ules. A CityGML module is an aggregate of normative aspects that must all be implemented as a whole by a conformant system. CityGML consists of a core module and thematic extension modules.

The CityGML core module defines the basic concepts and components of the CityGML data model. It is to be seen as the universal lower bound of the overall CityGML data model and a dependency of all thematic exten-sion modules. Thus, the core module is unique and must be implemented by any conformant system. Based on the CityGML core module, each extension module contains a logically separate thematic component of the CityGML data model. The extensions to the core are derived by vertically slicing the overall CityGML data model. Since the core module is contained (by reference) in each extension module, its general concepts and components are universal to all extension modules. The following thirteen thematic extension modules are introduced by version 2.0 of the CityGML standard. They are directly related to clauses of this document each covering the corresponding thematic field of CityGML:

- Appearance (cf. clause 9),
- Bridge (cf. clause 10.5)
- Building (cf. clause 10.3),
- CityFurniture (cf. clause 10.9),
- CityObjectGroup (cf. clause 10.11),
- Generics (cf. clause 10.12),
- LandUse (cf. clause 10.10),

- Relief (cf. clause 10.2),
- Transportation (cf. clause 10.7),
- Tunnel (cf. clause 10.4)
- Vegetation (cf. clause 10.8),
- WaterBody (cf. clause 10.6), and
- TexturedSurface [deprecated] (cf. clause 9.8).

The thematic decomposition of the CityGML data model allows for implementations to support any combination of extension modules in conjunction with the core module in order to be CityGML conformant. Thus, the exten-sion modules may be arbitrarily combined according to the information needs of an application or application domain. A combination of modules is called a CityGML profile. The union of all modules is defined as the CityGML base profile. The base profile is unique at any given time and forms the upper bound of the overall CityGML data model. Any other CityGML profile must be a valid subset of the base profile. By following the concept of CityGML modules and profiles, valid partial implementations of the CityGML data model may be realised in a well-defined way.

As for future development, each CityGML module may be further developed independently from other modules by expert groups and information communities. Resulting proposals and changes to modules may be introduced into future revisions of the CityGML standard without affecting the validity of other modules. Furthermore, thematic components not covered by the current CityGML data model may be added to future revisions of the standard by additional thematic extension modules. These additional extensions may establish dependency relations to any other existing CityGML module but shall at least be dependent on the CityGML core module. Consequently, the CityGML base profile may vary over time as new extensions are added. However, if a specific application has information needs to be modelled and exchanged which are beyond the scope of the CityGML data model, this application data can also be incorporated within the existing modules using CityGML's Applica-tion Domain Extension mechanism (cf. clause 10.13) or by employing the concepts of generic city objects and attributes (cf. chapter 10.12).

The introduced modularisation approach supports CityGML's versatility as a data modelling framework and exchange format addressing various application domains and different steps of 3D city modelling. For sake of clarity, applications should announce the level of conformance to the CityGML standard by declaring the em-ployed CityGML profile. Since the core module is part of all profiles, this should be realised by enumerating the implemented thematic extension modules. For example, if an implementation supports the Building module, the Relief module, and the Vegetation module in addition to the core, this should be announced by "CityGML [Building, Relief, Vegetation]". In case the base profile is supported, this should be indicated by "CityGML [full]".

9.1. CityGML core and extension modules

Each CityGML module is specified by its own XML Schema definition file and is defined within an individual and globally unique XML target namespace. According to dependency relations between modules, each module may, in addition, import namespaces associated to such related CityGML modules. However, a single namespace shall not be directly included in two modules. Thus, all elements belonging to one module are associ-ated to the module's namespace only. By this means, module elements are guaranteed to be properly separated and distinguishable in CityGML instance documents.

Compared to CityGML versions before 1.0, the aforementioned namespace conventions introduce an extra level of complexity to data files as there is no single CityGML namespace any more. In contrast, components of different CityGML modules and, thus, of different namespaces may be arbitrarily mixed within the same CityGML instance document. Furthermore, an application might have to parse instance documents containing elements of modules which are not employed by the application itself. These parsing problems though can easily be overcome by non-“schema-aware” applications, i.e. applications that do not parse and interpret GML applica-tion schemas in a generic way. Elements from different namespaces than those declared by the application’s employed CityGML profile could be skipped. Comparable observations have to be made when using CityGML’s Application Domain Extension mechanism (cf. clause 10.13).

As for version 2.0 of the CityGML standard, there are no two thematic extension modules related by dependen-cy. Thus, all extension modules are truly independent from each other and may be separately supported by implementations. However, the CityGML core module is a dependency for any extension module. This means that the XML schema file of the core module is imported by each XML schema file defining an extension.

The dependency relations between CityGML’s modules are illustrated in Fig. 8 using an UML package diagram. Each module is represented by a package. The package names correspond to the module names. A dashed arrow in the figure indicates that the schema at the tail of the arrow depends upon the schema at the head of the arrow. For CityGML modules, a dependency occurs where one schema <import>s another schema and accordingly the corresponding XML namespace. For example, the extension module Building imports the schema of the CityGML Core module. A short description of each module is given in Tab. 4.

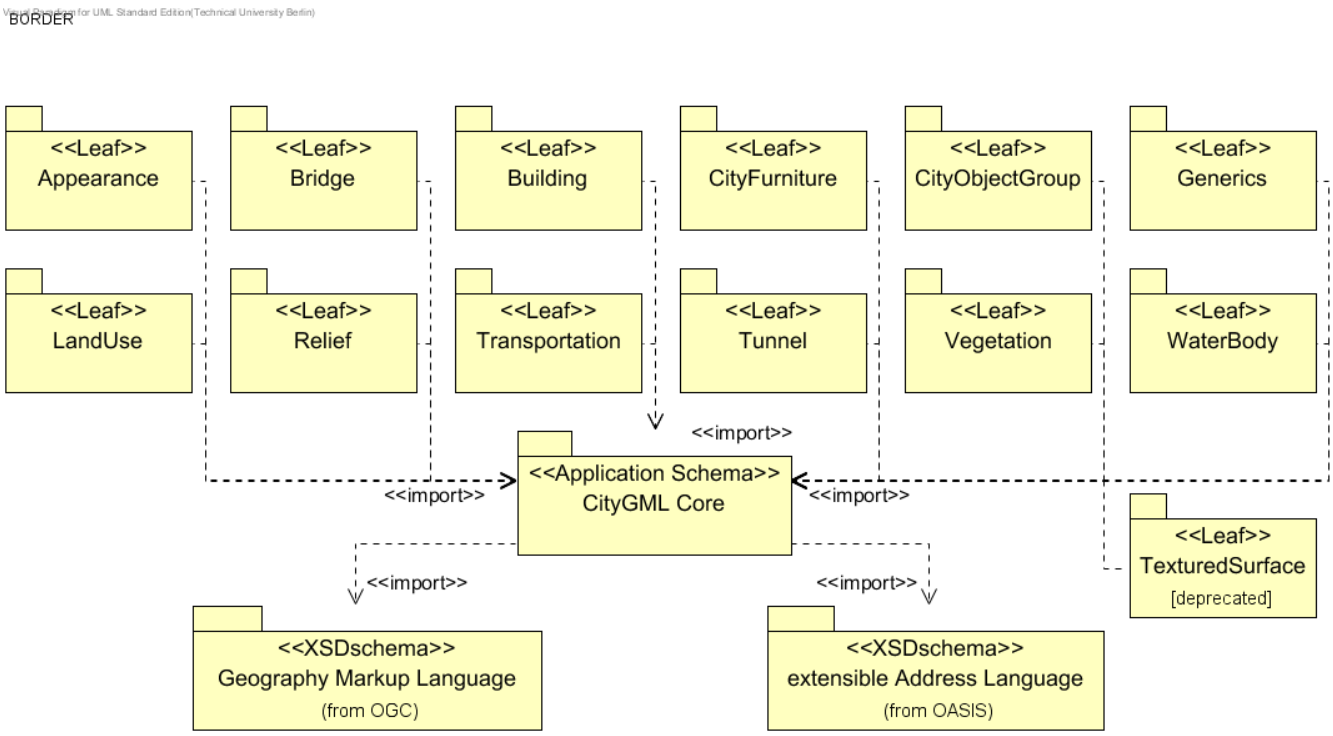


Figure 8. UML package diagram illustrating the separate modules of CityGML and their schema dependencies. Each extension module (indicated by the leaf packages) further imports the GML 3.1.1 schema definition in order to represent spatial properties of its thematic classes. For readability reasons, the corresponding dependencies have been omitted.

Module name	CityGML Core
XML namespace identifier	http://www.opengis.net/citygml/2.0
XML Schema file	cityGML.Base.xsd
Recommended namespace prefix	core
Module description	The CityGML Core module defines the basic components of the CityGML data model. Primarily, this comprises abstract base classes from which all thematic classes are (transitively) derived. But also non-abstract content common to more than one extension module, for example basic data types, is defined within the core module. The core module itself imports the XML schema definition files of GML version 3.1.1 and the OASIS extensible Address Language xAL.

Module name	Appearance
XML namespace identifier	http://www.opengis.net/citygml/appearance/2.0
XML Schema file	appearance.xsd

Recommended namespace prefix	app
Module description	The Appearance module provides the means to model appearances of CityGML features, i.e. observable properties of the feature’s surface. Appear-ance data may be stored for each city object. Therefore, the abstract base class _CityObject defined within the core module is augmented by an additional property using CityGML’s Application Domain Extension mechanism. Thus, the Appearance module has a deliberate impact on all thematic extension modules.

Module name	Bridge
XML namespace identifier	http://www.opengis.net/citygml/bridge/2.0
XML Schema file	bridge.xsd
Recommended namespace prefix	brid
Module description	The Bridge module allows the representation of thematic and spatial aspects of bridges, bridge parts, bridge installations, and interior bridge structures in four levels of detail (LOD 1 – 4).

Module name	Building
XML namespace identifier	http://www.opengis.net/citygml/building/2.0
XML Schema file	building.xsd
Recommended namespace prefix	bldg
Module description	The Building module allows the representation of thematic and spatial aspects of buildings, building parts, building installations, and interior building struc-tures in five levels of detail (LOD 0 – 4).

Module name	CityFurniture
XML namespace identifier	http://www.opengis.net/citygml/cityfurniture/2.0
XML Schema file	cityFurniture.xsd
Recommended namespace prefix	frn
Module description	The CityFurniture module is used to represent city furniture objects in cities. City furniture objects are immovable objects like lanterns, traffic signs, adver-tising columns, benches, or bus stops that can be found in traffic areas, residen-tial areas, on squares, or in built-up areas.

Module name	CityObjectGroup
XML namespace identifier	http://www.opengis.net/citygml/cityobjectgroup/2.0
XML Schema file	cityObjectGroup.xsd
Recommended namespace prefix	grp
Module description	The CityObjectGroup module provides a grouping concept for CityGML. Arbitrary city objects may be aggregated in groups according to user-defined criteria to represent and transfer these aggregations as part of the city model. A group may be further classified by specific attributes.

Module name	Generics
XML namespace identifier	http://www.opengis.net/citygml/generics/2.0
XML Schema file	generics.xsd
Recommended namespace prefix	gen
Module description	The Generics module provides generic extensions to the CityGML data model that may be used to model and exchange additional attributes and features not covered by the predefined thematic classes of CityGML. However, generic extensions shall only be used if appropriate thematic classes or attributes are not provided by any other CityGML module. In order to represent generic attributes, the Generics module augments the abstract base class _CityObject defined within the core module by an addition-al property using CityGML's Application Domain Extension mechanism. Thus, the Generics module has a deliberate impact on all thematic extension modules.

Module name	LandUse
XML namespace identifier	http://www.opengis.net/citygml/landuse/2.0
XML Schema file	landUse.xsd
Recommended namespace prefix	luse
Module description	The LandUse module allows for the representation of areas of the earth's surface dedicated to a specific land use.

Module name	Relief

XML namespace identifier	http://www.opengis.net/citygml/relief/2.0
XML Schema file	relief.xsd
Recommended namespace prefix	dem
Module description	The Relief module allows for the representation of the terrain in a city model. CityGML supports terrain representations in different levels of detail, reflect-ing different accuracies or resolutions. The terrain may be specified as a regular raster or grid, as a TIN, by break lines, and by mass points.

Module name	Transportation
XML namespace identifier	http://www.opengis.net/citygml/transportation/2.0
XML Schema file	transportation.xsd
Recommended namespace prefix	tran
Module description	The Transportation module is used to represent the transportation features within a city, for example roads, tracks, railways, or squares. Transportation features may be represented as a linear network or by geometrically describing their 3D surfaces.

Module name	Tunnel
XML namespace identifier	http://www.opengis.net/citygml/tunnel/2.0
XML Schema file	tunnel.xsd
Recommended namespace prefix	tun
Module description	The Tunnel module facilitates the representation of thematic and spatial aspects of tunnels, tunnel parts, tunnel installations, and interior tunnel struc-tures in four level of detail (LOD 1 – 4)

Module name	Vegetation
XML namespace identifier	http://www.opengis.net/citygml/vegetation/2.0
XML Schema file	vegetation.xsd
Recommended namespace prefix	veg

Module description	The Vegetation module provides thematic classes to represent vegetation objects. CityGML's vegetation model distinguishes between solitary vegeta-tion objects like trees, and vegetation areas which represent biotopes like forests or other plant communities.

Module name	WaterBody
XML namespace identifier	http://www.opengis.net/citygml/waterbody/2.0
XML Schema file	waterBody.xsd
Recommended namespace prefix	wtr
Module description	The WaterBody module represents the thematic aspects and 3D geometry of rivers, canals, lakes, and basins. It does, however, not inherit any hydrological or other dynamic aspects so far.

Module name	Textured Surface [deprecated]
XML namespace identifier	http://www.opengis.net/citygml/texturedsurface/2.0
XML Schema file	texturedSuface.xsd
Recommended namespace prefix	tex
Module description	The TexturedSurface module allows for assigning visual appearance properties (color, shininess, transparency) and textures to 3D surfaces. Due to inherent limitations of its modelling approach this module has been marked deprecated and is expected to be removed in future CityGML versions. Appearance information provided by this module can be converted to CityGML's Appear-ance module without information loss. Thus, the use of the TexturedSurface module is strongly discouraged.

9.2. CityGML profiles

A CityGML profile is a combination of thematic extension modules in conjunction with the core module of CityGML. Each CityGML instance document shall employ the CityGML profile appropriate to the provided data. In general, two approaches to employ a CityGML profile within an instance document can be differentiat-ed:

1. CityGML profile definition embedded inline the CityGML instance document A CityGML profile can be bound to an instance document using the schemaLocation attribute defined in the XML Schema instance namespace, http://www.w3.org/2001/XMLSchema-instance (commonly associated with the prefix xsi). The xsi:schemaLocation attribute provides a way to locate the XML Schema definition for namespaces defined in an XML instance document. Its value is a

whitespace-delimited list of pairs of Uniform Resource Identifiers (URIs) where each pair consists of a namespace followed by the location of that namespace's XML Schema definition, which is typically a .xsd file.

   By this means, the namespaces of the respective CityGML modules shall be defined within a CityGML instance document. The xsi:schemaLocation attribute then shall be used to provide the location to the respective XML Schema definition of each module. All example instance documents given in Annex G follow this first approach.

2. CityGML profile definition provided by a separate XML Schema definition file The CityGML profile may also be specified by its own XML Schema file. This schema file shall com-bine the appropriate CityGML modules by importing the corresponding XML Schema definitions. For this purpose, the import element defined in the XML Schema namespace shall be used, http://www.w3.org/2001/XMLSchema (commonly associated with the prefix xs). For the xs:import el-ement, the namespace of the imported CityGML module along with the location of the namespace's XML Schema definition have to be declared. In order to apply a CityGML profile to an instance docu-ment, the profile's schema has to be bound to the instance document using the xsi:schemaLocation at-tribute. The XML Schema file of the CityGML profile shall not contain any further content.

   The targetNamespace of the profile's schema shall differ from the namespaces of the imported CityGML modules. The namespace associated with the profile should be in control of the originator of the instance document and must be given as a previously unused and globally unique URI. The profile's XML Schema file must be available (or accessible on the internet) to everybody parsing the associated CityGML instance document.

The second approach is illustrated by the following example XML Schema definition for the base profile of CityGML. Since the base profile is the union of all CityGML modules, the corresponding XML Schema defini-tion imports each and every CityGML module. By this means, all components of the CityGML data model are available in and may be exchanged by instance documents referencing this example base profile. The schema definition file of the base profile is shipped with the CityGML schema package, and is accessible at http://schemas.opengis.net/citygml/profiles/base/2.0/CityGML.xsd.

NOTE replace XML with UML if feasible.

The following excerpt of a CityGML dataset exemplifies how to apply the base profile schema CityGML.xsd to a CityGML instance document. The dataset contains two building objects and a city object group. The base profile defined by CityGML.xsd is referenced using the xsi:schemaLocation attribute of the root element. Thus, all CityGML modules are employed by the instance document and no further references to the XML Schema documents of the CityGML modules are necessary.

NOTE replace XML with UML if feasible

Chapter 10. Spatial Model

NOTE This section was not generated from the CityGML Conceptual Model. TODO: identify where this info resides in the UML model, then update accordingly.

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-spatial	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

Spatial properties of CityGML features are represented by objects of GML3's geometry model. This model is based on the standard ISO 19107 'Spatial Schema' (Herring 2001), representing 3D geometry according to the well-known Boundary Representation (B-Rep, cf. Foley et al. 1995).

CityGML actually uses only a subset of the GML3 geometry package, defining a profile of GML3. This subset is depicted in Fig. 9 and Fig. 10. Further-more, GML3's explicit Boundary Representation is extended by scene graph concepts, which allow the represen-tation of the geometry of features with the same shape implicitly and thus more space efficiently (chapter 8.2).

NOTE Version 2.0 only provides conformance requirements for implicit geometries. Additional requirements will be needed for the other categories.

10.1. Geometric-topological model

NOTE short intro here

10.1.1. Primitives and Composits

NOTE short intro here

For a more detailed discussion of Primitives and Composits, see CityGML Best Practice Section nnn.

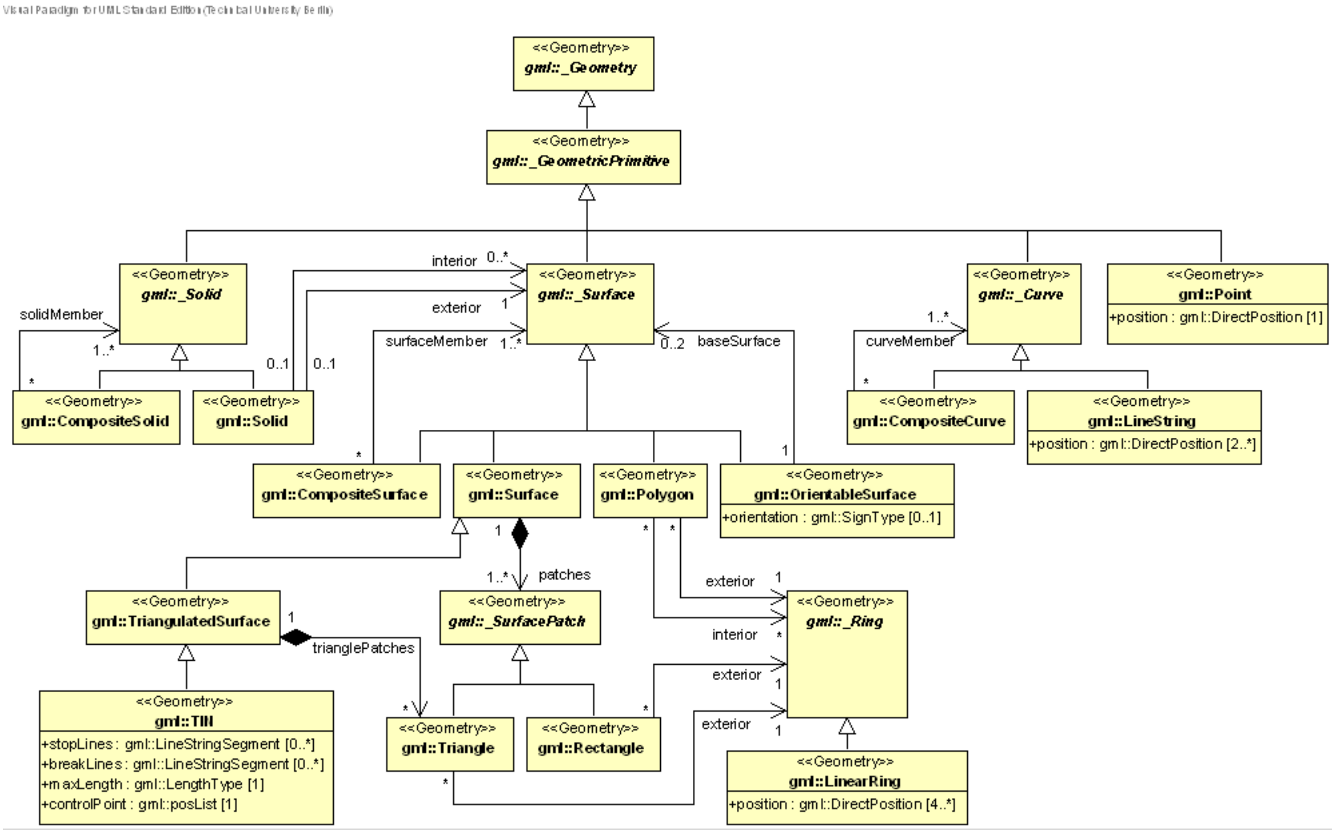


Figure 9. UML diagram of CityGML's geometry model (subset and profile of GML3): Primitives and Composites.

10.1.2. Complexes and Aggregates

NOTE	<u>short</u> intro here

For a more detailed discussion of Complexes and Aggregates, see [CityGML Best Practice Section nnn](#).

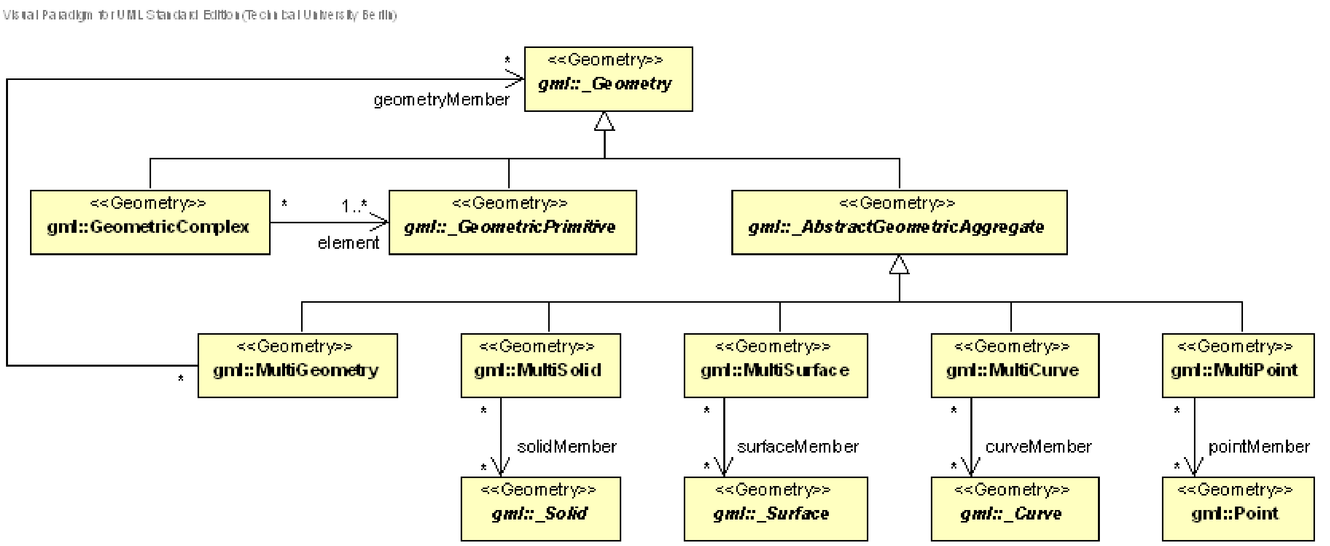


Figure 10. UML diagram of CityGML's geometry model: Complexes and Aggregates

10.1.3. Combined Geometries

NOTE	short intro here

For a more detailed discussion of Combined Geometries, see CityGML Best Practice Section nnn.

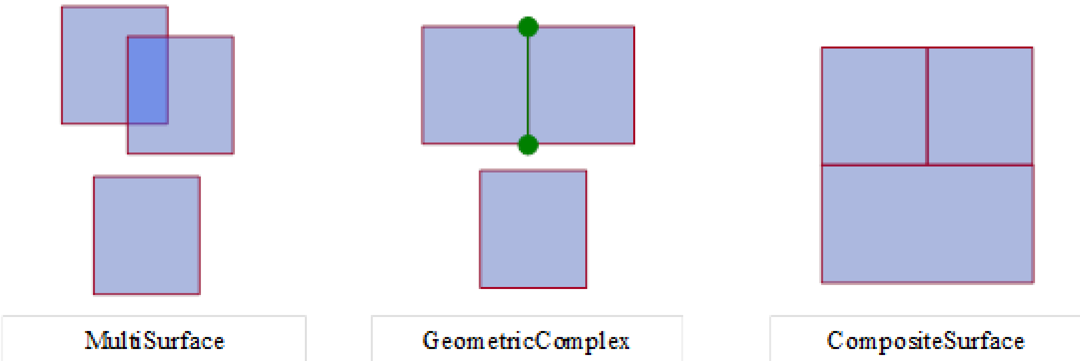


Figure 11. Combined geometries

10.1.4. Recursive Aggregation

NOTE	short intro here

For a more detailed discussion of Recursive Aggragation, see CityGML Best Practice Section nnn.

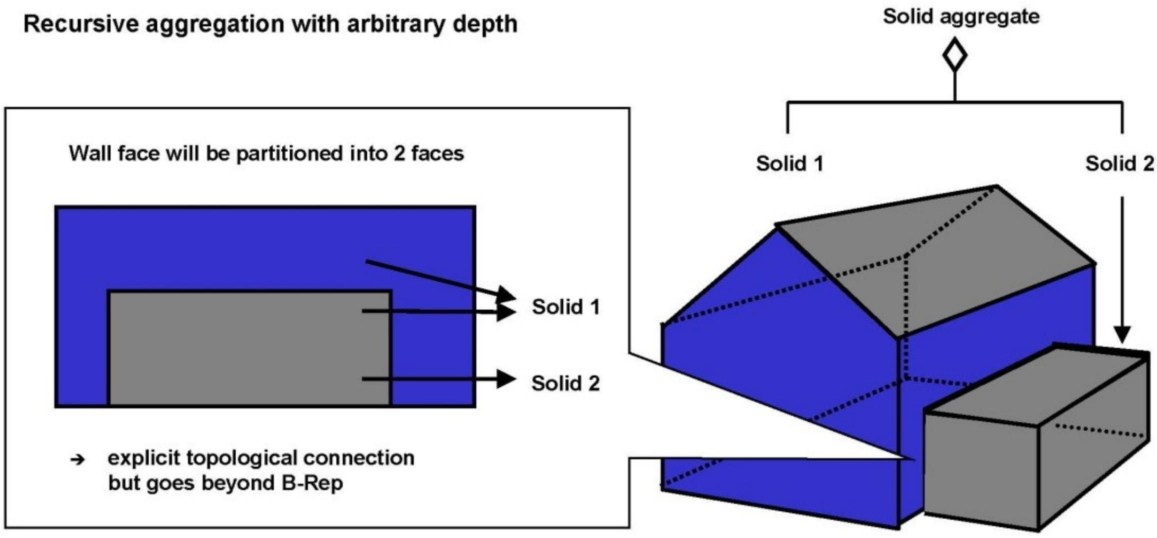


Figure 12. Recursive aggregation of objects and geometries in CityGML (graphic: IGG Uni Bonn).

10.2. Spatial Reference System

NOTE short intro here

For a more detailed discussion of Spatial Reference Systems, see CityGML Best Practice Section nnn.

NOTE add SRS requirements here

10.3. Implicit geometries, prototypic objects, scene graph concepts

NOTE short intro here

For a more detailed discussion of Implicit Geometries, see CityGML Best Practice Section nnn.

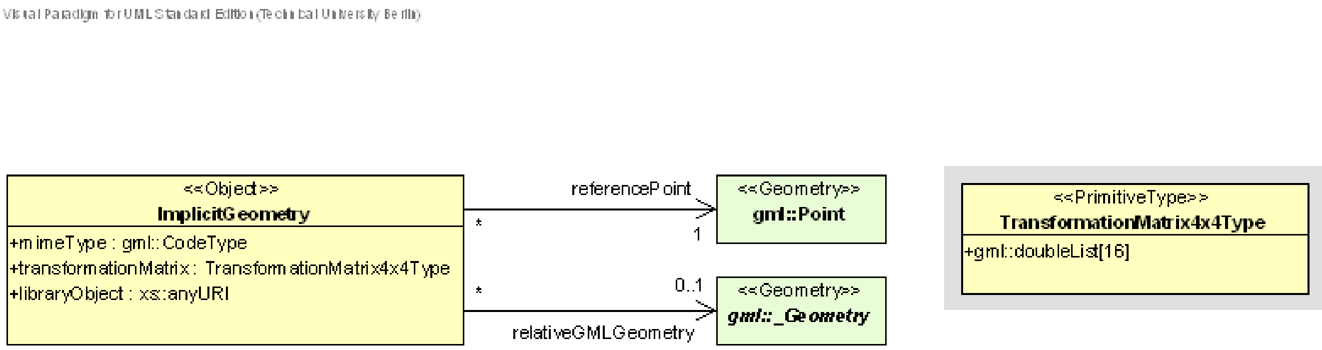


Figure 13. UML diagram of ImplicitGeometries. Prefixes are used to indicate XML namespaces associated with model elements. Element names without a prefix are defined within the CityGML Core module.

Requirement 1	**/req/spatial/base**
A	In order to geometrically represent a feature using the concept of implicit geometries, the corresponding thematic class of the feature shall define a spatial property of the type ImplicitRepresentationProperty-Type. Thus, for all CityGML extension modules only the type ImplicitRepresentationPropertyType shall be used for spatial properties providing implicit geometries.
B	If the shape of an implicit geometry is referenced by an URI using the libraryObject property (type: xs:anyURI) of the element ImplicitGeometry, also the MIME type of the denoted object must be speci-fied.

Requirement 2	**/req/spatial/refIntegrity**

A	The type ImplicitRepresentationPropertyType may contain an ImplicitGeometry element inline or an XLink reference to a remote ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the corresponding property of type ImplicitRepresentationPropertyType may only point to a remote ImplicitGeometry element (where remote ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

Chapter 11. Appearance Model

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-appearance	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

In addition to spatial properties, CityGML features have appearances – observable properties of the feature’s surface. Appearances are not limited to visual data but represent arbitrary categories called themes such as infrared radiation, noise pollution, or earthquake-induced structural stress. Each LOD can have an individual appearance for a specific theme. An appearance is composed of data for each surface geometry object, i.e. surface data. A single surface geometry object may have surface data for multiple themes. Similarly, surface data can be shared by multiple surface geometry objects (e.g. road paving). Finally, surface data values can either be constant across a surface or depend on the exact location within the surface.

CityGML’s appearance model is defined within the extension module Appearance (cf. chapter 7). The UML diagram of the appearance model is illustrated in Appearance UML Diagram. The Data Dictionary for the Appearance Package is provided in section Appearance Data Dictionary.

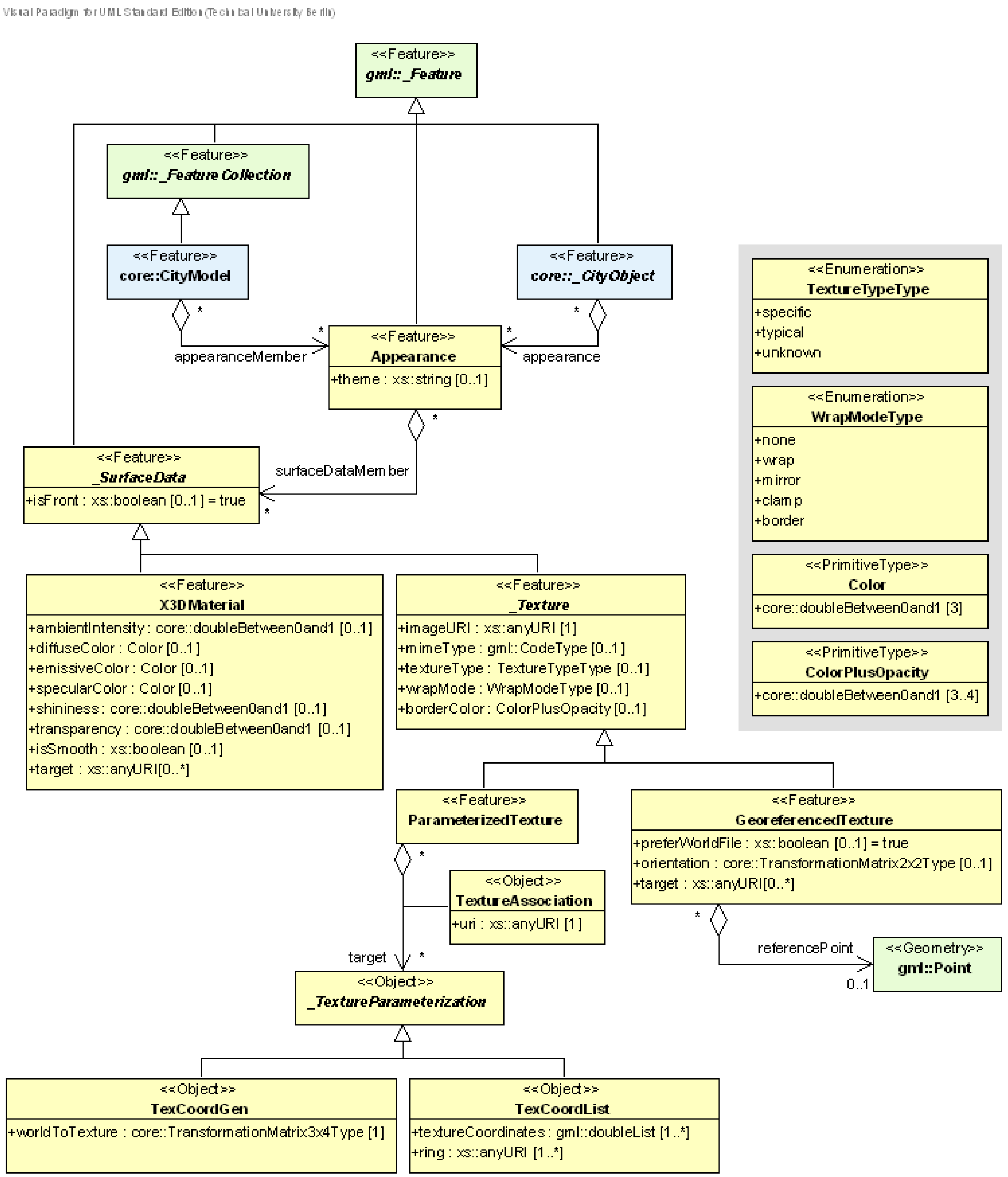


Figure 14. UML diagram of CityGML's appearance model.

The UML diagram of CityGML's appearance model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

11.1. Appearance Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

11.1.1. Class AbstractSurfaceData

Subclass of <←section,>>

Class AbstractSurfaceData
Definition: Subtype Of: <←section,>> StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: X3DMaterial Target Role: Target Class: AbstractSurfaceData
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractTexture Target Role: Target Class: AbstractSurfaceData
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Appearance Target Role: surfaceData Target Class: AbstractSurfaceData
Attributes
Attribute Name: isFront Value Type: Boolean Definition: SIG3D: Indicates whether the X3DMaterial, GeoreferencedTexture or, ParametrizedTexture is assigned to the front side or back side of the surface Multiplicity: [0..1] Stereotype: «Property»

11.1.2. Class AbstractTexture

Subclass of AbstractSurfaceData

Class AbstractTexture	
Definition:	
Subtype Of:	AbstractSurfaceData
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTexture
Target Role:	
Target Class:	AbstractSurfaceData
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GeoreferencedTexture
Target Role:	
Target Class:	AbstractTexture
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	ParameterizedTexture
Target Role:	
Target Class:	AbstractTexture
Attributes	
Attribute Name:	borderColor
Value Type:	ColorPlusOpacity
Definition:	SIG3D: Color definition for texture borders.
Multiplicity:	[0..1]
Stereotype:	«Property»
Attribute Name:	imageURI
Value Type:	URI
Definition:	SIG3D: URI identifying the image data resource for the texture
Multiplicity:	
Stereotype:	«Property»

Attribute Name: mimeType Value Type: Code Definition: SIG3D: Mime type describing the data format of the image resource Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: textureType Value Type: TextureType Definition: "SIG3D: Distinction between prototypical (value ""typical""), object specific (value ""specific"") textures, and textures with unknown classification (value ""unknown"")." Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: wrapMode Value Type: WrapMode Definition: Collada: Definition of behavior when accessing the texture outside the underlying image raster Multiplicity: [0..1] Stereotype: «Property»

11.1.3. Class Appearance

Subclass of AbstractAppearance

Class Appearance
Definition: Subtype Of: AbstractAppearance StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Appearance Target Role: Target Class: AbstractAppearance
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Appearance Target Role: surfaceData Target Class: AbstractSurfaceData
Attributes

Attribute Name: theme Value Type: CharacterString Definition: SIG3D: Theme name for all surfaceDataMembers. A theme is a catagory defining the semantics of the referenced surfaceDataMembers (e.g. infrared radiation, Multiplicity: [0..1] Stereotype: «Property»

11.1.4. Class Color

Subclass of DoubleBetween0and1List

Class Color
Definition: Subtype Of: DoubleBetween0and1List StereoType: «BasicType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Color Target Role: Target Class: DoubleBetween0and1List
Name: Type: NoteLink Direction: Source Role: Source Class: Note Target Role: Target Class: Color
Attributes

11.1.5. Class ColorPlusOpacity

Subclass of DoubleBetween0and1List

Class ColorPlusOpacity
Definition: Subtype Of: DoubleBetween0and1List StereoType: «BasicType»
Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	ColorPlusOpacity
Target Role:	
Target Class:	DoubleBetween0and1List
Name:	
Type:	NoteLink
Direction:	
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	ColorPlusOpacity
Attributes	

11.1.6. Class GeoreferencedTexture

Subclass of AbstractTexture

Class GeoreferencedTexture	
Definition:	
Subtype Of:	AbstractTexture
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	GeoreferencedTexture
Target Role:	referencePoint
Target Class:	GM_Point
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GeoreferencedTexture
Target Role:	
Target Class:	AbstractTexture
Attributes	

Attribute Name: orientation Value Type: TransformationMatrix2x2 Definition: SIG3D: Defines the rotation and scaling of the image in form of a 2x2 matrix (a list of 4 doubles in row-major order). Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: preferWorldFile Value Type: Boolean Definition: SIG3D: Flag for defining the precedence of an accompanying worldfile before the georeference included in the image source. If this value is false, the world file Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: target Value Type: URI Definition: SIG3D: Geometry object which is associated with the texture. Multiplicity: [0..*] Stereotype: «Property»

11.1.7. Class ParameterizedTexture

Subclass of AbstractTexture

Class ParameterizedTexture
Definition: Subtype Of: AbstractTexture StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: ParameterizedTexture Target Role: Target Class: AbstractTexture
Name: Type: AssociationClass Direction: Source → Destination Source Role: Source Class: ParameterizedTexture Target Role: textureParameterization Target Class: AbstractTextureParameterization
Attributes

11.1.8. Class TextureAssociation

Subclass of <←section,>>

Class TextureAssociation
Definition: Subtype Of: <←section,>> StereoType: «ObjectType»
Associations
Attributes
Attribute Name: uri Value Type: URI Definition: SIG3D: Link to the geometry object to be textured. Multiplicity: Stereotype: «Property»

11.1.9. Class X3DMaterial

Subclass of AbstractSurfaceData

Class X3DMaterial
Definition: Subtype Of: AbstractSurfaceData StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: X3DMaterial Target Role: Target Class: AbstractSurfaceData
Attributes
Attribute Name: ambientIntensity Value Type: DoubleBetween0and1 Definition: X3D: Minimum percentage of diffuseColor that is visible regardless of light sources Multiplicity: [0..1] Stereotype: «Property»

Attribute Name: diffuseColor Value Type: Color Definition: X3D: Color of the diffusely reflected light Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: emissiveColor Value Type: Color Definition: X3D: Color of the light emitted by the surface Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: isSmooth Value Type: Boolean Definition: X3D: Hint for normal interpolation. If true vertex normals used for shading. Otherwise normals are constant for the patch. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: shininess Value Type: DoubleBetween0and1 Definition: X3D: Controls the sharpness of specular highlights. 0 produces a soft glow. 1 produces sharp highlights. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: specularColor Value Type: Color Definition: X3D: Color of the directly reflected light Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: target Value Type: URI Definition: X3D: URI identifying the target surface geometry to apply the material properties Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: transparency Value Type: DoubleBetween0and1 Definition: X3D: Degree of transparency of the surface. 0 means fully opaque. 1 means fully transparent. Multiplicity: [0..1] Stereotype: «Property»

11.1.10. Class AbstractTextureParameterization

Subclass of <←section,>>

Class AbstractTextureParameterization

Definition: Subtype Of: <←section,>> StereoType: «DataType»

Associations	
Name: Type: Direction: Source Role: Source Class: Target Role: Target Class:	 Generalization Source → Destination TexCoordGen AbstractTextureParameterization
Name: Type: Direction: Source Role: Source Class: Target Role: Target Class:	 AssociationClass Source → Destination ParameterizedTexture textureParameterization AbstractTextureParameterization
Name: Type: Direction: Source Role: Source Class: Target Role: Target Class:	 Generalization Source → Destination TexCoordList AbstractTextureParameterization

Attributes

11.1.11. Class TexCoordGen

Subclass of AbstractTextureParameterization

Class TexCoordGen
Definition: Subtype Of: <←section,>> StereoType: «DataType»

Associations	
Name: Type: Direction: Source Role: Source Class: Target Role: Target Class:	 Generalization Source → Destination TexCoordGen AbstractTextureParameterization

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: TexCoordGen
Target Role: crs
Target Class: SC_CRS

Attributes

Attribute Name: worldToTexture
Value Type: TransformationMatrix3x4
Definition: SIG3D: 3x4 Matrix that defines the transformation between world coordinates and texture coordinates.
Multiplicity:
Stereotype: «Property»

11.1.12. Class TexCoordList

Subclass of AbstractTextureParameterization

Class TexCoordList

Definition:
Subtype Of: <←section,>>
StereoType: «DataType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: TexCoordList
Target Role:
Target Class: AbstractTextureParameterization

Attributes

Attribute Name: ring
Value Type: URI
Definition: SIG3D: gml:ids of the LinearRings that are parameterized using the given texture coordinates
Multiplicity: [1..*]
Stereotype: «Property»

Attribute Name: textureCoordinates
Value Type: DoubleList
Definition: SIG3D: List of texture coordinates with two doubles per ring vertex
Multiplicity: [1..*]
Stereotype: «Property»

11.1.13. Class TextureType

Subclass of <←section,>>

Class TextureType
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes
Attribute Name: specific Value Type: Definition: SIG3D: The texture is specific for a certain object Multiplicity: Stereotype:
Attribute Name: typical Value Type: Definition: SIG3D: The texture is typical for a kind of object Multiplicity: Stereotype:
Attribute Name: unknown Value Type: Definition: SIG3D: The texture type is unknown. Multiplicity: Stereotype:

11.1.14. Class WrapMode

Subclass of <←section,>>

Class WrapMode
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes
Attribute Name: none Value Type: Definition: COLLADA: The resulting color is fully transparent Multiplicity: Stereotype:

Attribute Name:	wrap
Value Type:	
Definition:	COLLADA: The texture is repeated
Multiplicity:	
Stereotype:	
Attribute Name:	mirror
Value Type:	
Definition:	COLLADA: The texture is repeated and mirrored
Multiplicity:	
Stereotype:	
Attribute Name:	clamp
Value Type:	
Definition:	COLLADA: The texture is clamped to its edges
Multiplicity:	
Stereotype:	
Attribute Name:	border
Value Type:	
Definition:	COLLADA: The resulting color is specified by the borderColor element (RGBA)
Multiplicity:	
Stereotype:	

11.1.15. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

11.1.16. Requirements

Requirement 3	**/req/appearance/base**
A	A surface geometry object may be the target of at most two textures and two materials (one for front and back respectively) per theme.
B	The referencePoint property (type: gml:PointPropertyType) of the element GeoreferencedTexture may only contain or reference a point geometry object with 2D coordinate values.

C	Texture coordinates given by the textureCoordinates property of the element TexCoordList define an explicit mapping of a surface’s boundary points to points in texture space. Each boundary point of the surface must receive a corresponding coordinate pair in texture space. The coordinate pair in texture space shall be given as two doubles per boundary point. The order of the coordinate pairs must follow the order of the boundary points in the CityGML document (regardless of a possibly flipped surface orientation). Each gml:LinearRing composing the boundary of the target surface geometry object re-quires its own set of texture coordinates.
D	A GeoreferencedTexture element must provide either internal or external georeference, otherwise it is invalid. Internal georeference shall be declared by the referencePoint property (type: gml:PointPropertyType) and the orientation property (type: core:TransformationMatrix2x2Type) of the element GeoreferencedTexture. External georeference may be provided by the texture image file itself (e.g. GeoTIFF) or by an accompanying world file.

Requirement 4	**/req/appearance/refIntegrity**
A	The appearanceMember element (type: AppearancePropertyType) may contain an Appearance element inline or an XLink reference to a remote Appearance element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the appearanceMember element may only point to a remote Appearance element (where remote Appearance elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
B	The appearance property (type: AppearancePropertyType) of the element core:_CityObject may con-tain an Appearance element inline or an XLink reference to a remote Appearance element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the appearance property may only point to a remote Appearance element (where remote Appearance elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

C	The surfaceDataMember property (type: SurfaceDataPropertyType) of the element Appearance may contain a _SurfaceData element inline or an XLink reference to a remote _SurfaceData element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the surfaceDataMember property may only point to a remote _SurfaceData element (where remote _SurfaceData elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
D	The target property (type: TextureAssociationType) of the element ParameterizedTexture may contain a _TextureParameterization element inline or an XLink reference to a remote _ TextureParameterization element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the target property may only point to a remote _ TextureParameterization element (where remote _ TexturePa-rameterization elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
E	The target property (type xs:anyURI) of the element GeoreferencedTexture shall specify the gml:id of the target surface geometry object which may only be of type gml:AbstractSurfaceType or gml:MultiSurface.
F	The uri attribute of the complex type TextureAssociationType shall specify the gml:id of the target sur-face geometry object which may only be of type gml:AbstractSurfaceType or gml:MultiSurface.
G	The ring attribute of the textureCoordinates property of the element TexCoordList shall specify the gml:id of the target surface geometry object which may only be of type gml:LinearRing.
H	The target property (type xs:anyURI) of the element X3DMaterial shall specify the gml:id of the target surface geometry object which may only be of type gml:AbstractSurfaceType or gml:MultiSurface.

Chapter 12. CityGML Conceptual Model

This section provides a detailed discussion of the CityGML Conceptual Model.

12.1. Core

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-core	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

The CityGML Core module defines the basic concepts and components of the overall CityGML data model. It forms the universal lower bound of the CityGML data model and, thus, is a dependency of all extension modules. Consequently, the core module has to be implemented by any conformant system. Primarily, the core module provides the abstract base classes from which thematic classes within extension modules are (transitively) derived. Besides abstract type definitions, the core also contains non-abstract content, for example basic data types and thematic classes that may be used by more than one extension module. The UML diagram in Fig. 21 illustrates CityGML's core module, for the XML Schema definition see below and annex A.1.

The UML diagram of the CityGML Core is depicted in Core UML Diagram. The Data Dictionary for the Core Package is provided in section Core Data Dictionary.

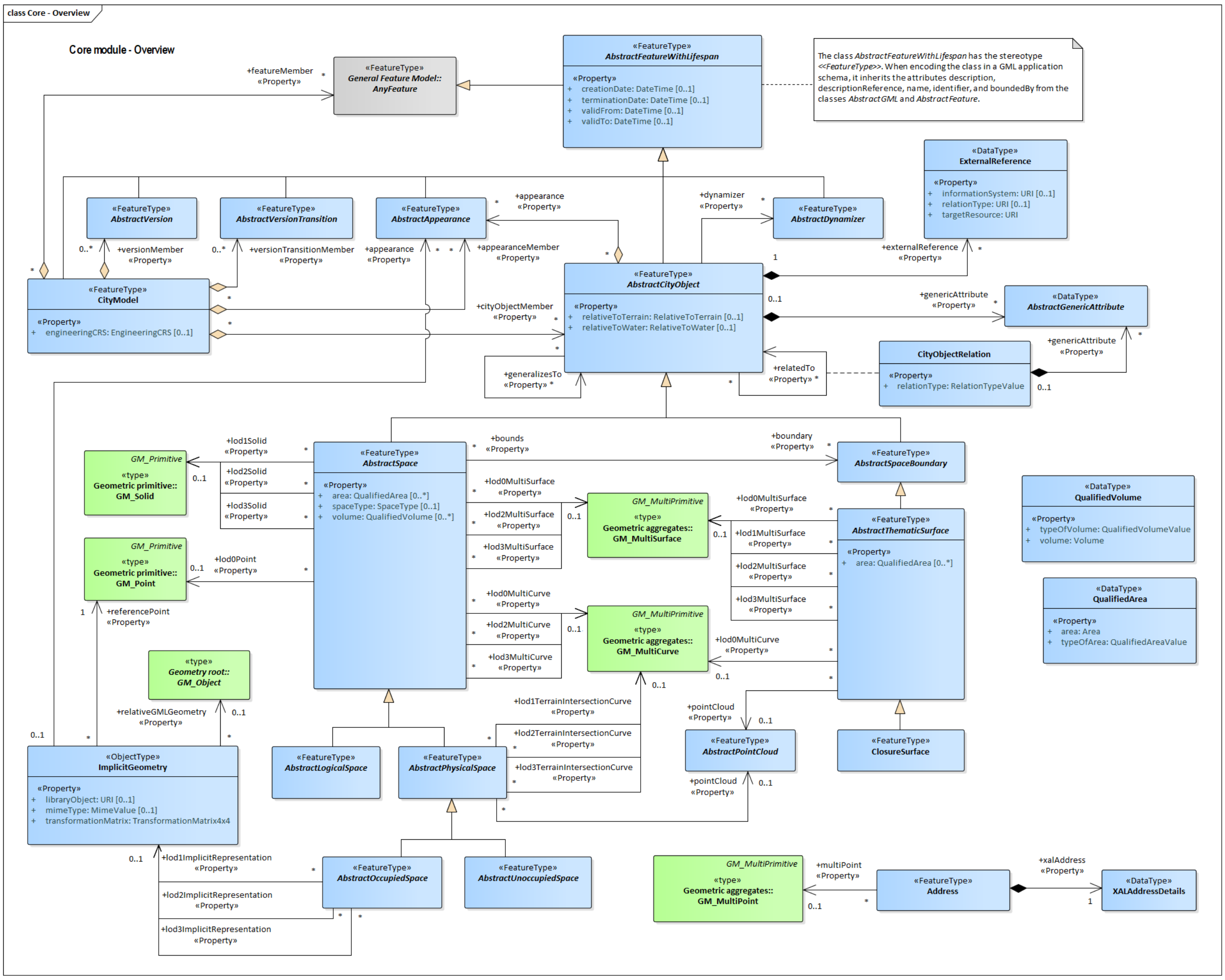


Figure 15. UML diagram of CityGML’s core module.

The UML diagram of CityGML’s core module. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.2. Core Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.2.1. Class AbstractAppearance

Subclass of AbstractFeatureWithLifespan

Class AbstractAppearance

Definition: Subtype Of: AbstractFeatureWithLifespan StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractAppearance Target Role: Target Class: AbstractFeatureWithLifespan
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractCityObject Target Role: appearance Target Class: AbstractAppearance
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityModel Target Role: appearanceMember Target Class: AbstractAppearance
Name: Type: Association Direction: Source → Destination Source Role: Source Class: ImplicitGeometry Target Role: appearance Target Class: AbstractAppearance
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Appearance Target Role: Target Class: AbstractAppearance
Attributes

12.2.2. Class AbstractCityObject

Subclass of AbstractFeatureWithLifespan

Class AbstractCityObject	
Definition:	
Subtype Of:	AbstractFeatureWithLifespan
StereoType:	«FeatureType»

Associations

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	appearance
Target Class:	AbstractAppearance

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	externalReference
Target Class:	ExternalReference

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	dynamizer
Target Class:	AbstractDynamizer

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	generalizesTo
Target Class:	AbstractCityObject

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	genericAttribute
Target Class:	AbstractGenericAttribute

Name:	
Type:	AssociationClass
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	relatedTo
Target Class:	AbstractCityObject

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CityModel
Target Role:	cityObjectMember
Target Class:	AbstractCityObject

Name:	
Type:	AssociationClass
Direction:	Source → Destination
Source Role:	
Source Class:	CityObjectGroup
Target Role:	groupMember
Target Class:	AbstractCityObject

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	SensorConnection
Target Role:	sensorLocation
Target Class:	AbstractCityObject

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CityObjectGroup
Target Role:	parent
Target Class:	AbstractCityObject

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: AbstractSpaceBoundary
Target Role:
Target Class: AbstractCityObject

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: AbstractCityObject
Target Role: generalizesTo
Target Class: AbstractCityObject

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: AbstractSpace
Target Role:
Target Class: AbstractCityObject

Name:
Type: AssociationClass
Direction: Source → Destination
Source Role:
Source Class: AbstractCityObject
Target Role: relatedTo
Target Class: AbstractCityObject

Attributes

Attribute Name: relativeToTerrain
Value Type: RelativeToTerrain
Definition: SIG3D: Vertical position of the AbstractCityObject relative to the surrounding terrain.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: relativeToWater
Value Type: RelativeToWater
Definition: SIG3D: Vertical position of the AbstractCityObject relative to a surrounding water surface.
Multiplicity: [0..1]
Stereotype: «Property»

12.2.3. Class AbstractDynamizer

Subclass of AbstractFeatureWithLifespan

Class AbstractDynamizer	
Definition:	
Subtype Of:	AbstractFeatureWithLifespan
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractDynamizer
Target Role:	
Target Class:	AbstractFeatureWithLifespan
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	dynamizer
Target Class:	AbstractDynamizer
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Dynamizer
Target Role:	
Target Class:	AbstractDynamizer
Attributes	

12.2.4. Class AbstractFeatureWithLifespan

Subclass of AnyFeature

Class AbstractFeatureWithLifespan	
Definition:	
Subtype Of:	AnyFeature
StereoType:	«FeatureType»
Associations	

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractFeatureWithLifespan
Target Role:	
Target Class:	AnyFeature

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractVersion
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Transaction
Target Role:	oldFeature
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractAppearance
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Version
Target Role:	versionMember
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractVersionTransition
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	CityModel
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Transaction
Target Role:	newFeature
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractDynamizer
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractCityObject
Target Role:	
Target Class:	AbstractFeatureWithLifespan

Attributes

Attribute Name:	creationDate
Value Type:	DateTime
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name: terminationDate Value Type: DateTime Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: validFrom Value Type: DateTime Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: validTo Value Type: DateTime Definition: Multiplicity: [0..1] Stereotype: «Property»

12.2.5. Class AbstractLogicalSpace

Subclass of AbstractSpace

Class AbstractLogicalSpace
Definition: Subtype Of: AbstractSpace StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractLogicalSpace Target Role: Target Class: AbstractSpace
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: CityObjectGroup Target Role: Target Class: AbstractLogicalSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GenericLogicalSpace
Target Role:	
Target Class:	AbstractLogicalSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	
Target Class:	AbstractLogicalSpace
Attributes	

12.2.6. Class AbstractOccupiedSpace

Subclass of AbstractPhysicalSpace

Class AbstractOccupiedSpace	
Definition:	
Subtype Of:	AbstractPhysicalSpace
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractOccupiedSpace
Target Role:	lod2ImplicitRepresentation
Target Class:	ImplicitGeometry
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractOccupiedSpace
Target Role:	
Target Class:	AbstractPhysicalSpace

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractOccupiedSpace
Target Role:	lod1ImplicitRepresentation
Target Class:	ImplicitGeometry

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractOccupiedSpace
Target Role:	lod3ImplicitRepresentation
Target Class:	ImplicitGeometry

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractFurniture
Target Role:	
Target Class:	AbstractOccupiedSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	WaterBody
Target Role:	
Target Class:	AbstractOccupiedSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GenericOccupiedSpace
Target Role:	
Target Class:	AbstractOccupiedSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstructiveElement
Target Role:	
Target Class:	AbstractOccupiedSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractVegetationObject
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractInstallation
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	CityFurniture
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstruction
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractFillingElement
Target Role:	
Target Class:	AbstractOccupiedSpace
Attributes	

12.2.7. Class AbstractPhysicalSpace

Subclass of AbstractSpace

Class AbstractPhysicalSpace

Definition:	
Subtype Of:	AbstractSpace
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	
Target Class:	AbstractSpace
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	pointCloud
Target Class:	AbstractPointCloud
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	lod3TerrainIntersectionCurve
Target Class:	GM_MultiCurve
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	lod2TerrainIntersectionCurve
Target Class:	GM_MultiCurve
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	lod1TerrainIntersectionCurve
Target Class:	GM_MultiCurve

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractUnoccupiedSpace
Target Role:	
Target Class:	AbstractPhysicalSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractOccupiedSpace
Target Role:	
Target Class:	AbstractPhysicalSpace
Attributes	

12.2.8. Class AbstractPointCloud

Subclass of <←section,>>

Class AbstractPointCloud	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	PointCloud
Target Role:	
Target Class:	AbstractPointCloud
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	MassPointRelief
Target Role:	pointCloud
Target Class:	AbstractPointCloud

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	pointCloud
Target Class:	AbstractPointCloud
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	pointCloud
Target Class:	AbstractPointCloud
Attributes	

12.2.9. Class AbstractSpace

Subclass of AbstractCityObject

Class AbstractSpace	
Definition:	
Subtype Of:	AbstractCityObject
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod3MultiCurve
Target Class:	GM_MultiCurve
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod2MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod3Solid
Target Class:	GM_Solid

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod2Solid
Target Class:	GM_Solid

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod1Solid
Target Class:	GM_Solid

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod0Point
Target Class:	GM_Point

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	bounds
Source Class:	AbstractSpace
Target Role:	boundary
Target Class:	AbstractSpaceBoundary

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod0MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod2MultiCurve
Target Class:	GM_MultiCurve

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod3MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	lod0MultiCurve
Target Class:	GM_MultiCurve

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractSpace
Target Role:	
Target Class:	AbstractCityObject

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractPhysicalSpace
Target Role:	
Target Class:	AbstractSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractLogicalSpace
Target Role:	
Target Class:	AbstractSpace

Attributes

Attribute Name: area Value Type: QualifiedArea Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: spaceType Value Type: SpaceType Definition: Degree of openness of an abstract space. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: volume Value Type: QualifiedVolume Definition: Multiplicity: [0..*] Stereotype: «Property»

12.2.10. Class AbstractSpaceBoundary

Subclass of AbstractCityObject

Class AbstractSpaceBoundary
Definition: Subtype Of: AbstractCityObject StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractSpaceBoundary Target Role: Target Class: AbstractCityObject
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractThematicSurface Target Role: Target Class: AbstractSpaceBoundary

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	bounds
Source Class:	AbstractSpace
Target Role:	boundary
Target Class:	AbstractSpaceBoundary
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	ReliefFeature
Target Role:	
Target Class:	AbstractSpaceBoundary
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractReliefComponent
Target Role:	
Target Class:	AbstractSpaceBoundary
Attributes	

12.2.11. Class AbstractThematicSurface

Subclass of AbstractSpaceBoundary

Class AbstractThematicSurface	
Definition:	
Subtype Of:	AbstractSpaceBoundary
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	lod1MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	
Target Class:	AbstractSpaceBoundary

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	lod0MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	lod2MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	lod3MultiSurface
Target Class:	GM_MultiSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	lod0MultiCurve
Target Class:	GM_MultiCurve

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractThematicSurface
Target Role:	pointCloud
Target Class:	AbstractPointCloud

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TrafficArea
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractInstallation
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstructionSurface
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Storey
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractWaterBoundarySurface
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Marking
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstruction
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	LandUse
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstructiveElement
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	HoleSurface
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractFillingSurface
Target Role:	
Target Class:	AbstractThematicSurface
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Hole
Target Role:	boundary
Target Class:	AbstractThematicSurface
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	HollowSpace
Target Role:	boundary
Target Class:	AbstractThematicSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	ClosureSurface
Target Role:	
Target Class:	AbstractThematicSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GenericThematicSurface
Target Role:	
Target Class:	AbstractThematicSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AuxiliaryTrafficArea
Target Role:	
Target Class:	AbstractThematicSurface
Attributes	

Attribute Name:	area
Value Type:	QualifiedArea
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

12.2.12. Class AbstractUnoccupiedSpace

Subclass of AbstractPhysicalSpace

Class AbstractUnoccupiedSpace	
Definition:	
Subtype Of:	AbstractPhysicalSpace
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractUnoccupiedSpace
Target Role:	
Target Class:	AbstractPhysicalSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GenericUnoccupiedSpace
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTransportationSpace
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	
Target Class:	AbstractUnoccupiedSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TrafficSpace
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AuxiliaryTrafficSpace
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Hole
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	HollowSpace
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	ClearanceSpace
Target Role:	
Target Class:	AbstractUnoccupiedSpace
Attributes	

12.2.13. Class AbstractVersion

Subclass of AbstractFeatureWithLifespan

Class AbstractVersion	
Definition:	
Subtype Of:	AbstractFeatureWithLifespan
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractVersion
Target Role:	
Target Class:	AbstractFeatureWithLifespan
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CityModel
Target Role:	versionMember
Target Class:	AbstractVersion
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Version
Target Role:	
Target Class:	AbstractVersion

Attributes

12.2.14. Class AbstractVersionTransition

Subclass of AbstractFeatureWithLifespan

Class AbstractVersionTransition	
Definition:	
Subtype Of:	AbstractFeatureWithLifespan
StereoType:	«FeatureType»

Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractVersionTransition
Target Role:	
Target Class:	AbstractFeatureWithLifespan
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CityModel
Target Role:	versionTransitionMember
Target Class:	AbstractVersionTransition
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	VersionTransition
Target Role:	
Target Class:	AbstractVersionTransition
Attributes	

12.2.15. Class Address

Subclass of <←section,>>

Class Address	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Address
Target Role:	xalAddress
Target Class:	XALAddressDetails

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Address
Target Role:	multiPoint
Target Class:	GM_MultiPoint
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	DoorSurface
Target Role:	address
Target Class:	Address
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBridge
Target Role:	address
Target Class:	Address
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	address
Target Class:	Address
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingUnit
Target Role:	address
Target Class:	Address
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Door
Target Role:	address
Target Class:	Address

Attributes

12.2.16. Class CityModel

Subclass of AbstractFeatureWithLifespan

Class CityModel
Definition: Subtype Of: AbstractFeatureWithLifespan StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityModel Target Role: cityObjectMember Target Class: AbstractCityObject
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityModel Target Role: versionMember Target Class: AbstractVersion
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityModel Target Role: versionTransitionMember Target Class: AbstractVersionTransition
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityModel Target Role: appearanceMember Target Class: AbstractAppearance
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityModel Target Role: featureMember Target Class: AnyFeature

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: CityModel
Target Role:
Target Class: AbstractFeatureWithLifespan

Attributes

Attribute Name: engineeringCRS
Value Type: EngineeringCRS
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

12.2.17. Class CityObjectRelation

Subclass of <←section,>>

Class CityObjectRelation

Definition:
Subtype Of: <←section,>>
StereoType:

Associations

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: CityObjectRelation
Target Role: genericAttribute
Target Class: AbstractGenericAttribute

Attributes

Attribute Name: relationType
Value Type: RelationTypeValue
Definition:
Multiplicity:
Stereotype: «Property»

12.2.18. Class ClosureSurface

Subclass of AbstractThematicSurface

Class ClosureSurface

Definition: Subtype Of: AbstractThematicSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: ClosureSurface Target Role: Target Class: AbstractThematicSurface
Attributes

12.2.19. Class DoubleBetween0and1

Subclass of <←section,>>

Class DoubleBetween0and1
Definition: Subtype Of: <←section,>> StereoType: «BasicType»
Associations
Name: Type: NoteLink Direction: Source Role: Source Class: Note Target Role: Target Class: DoubleBetween0and1
Attributes

12.2.20. Class DoubleBetween0and1List

Subclass of <←section,>>

Class DoubleBetween0and1List
Definition: Subtype Of: <←section,>> StereoType: «BasicType»
Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Color Target Role: Target Class: DoubleBetween0and1List
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: ColorPlusOpacity Target Role: Target Class: DoubleBetween0and1List
Attributes
Attribute Name: list Value Type: DoubleBetween0and1 Definition: Multiplicity: Stereotype:

12.2.21. Class DoubleList

Subclass of <←section,>>

Class DoubleList
Definition: Subtype Of: <←section,>> StereoType: «BasicType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TransformationMatrix4x4 Target Role: Target Class: DoubleList
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TransformationMatrix3x4 Target Role: Target Class: DoubleList

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TransformationMatrix2x2
Target Role:	
Target Class:	DoubleList
Attributes	
Attribute Name:	list
Value Type:	Real
Definition:	
Multiplicity:	
Stereotype:	

12.2.22. Class ImplicitGeometry

Subclass of <←section,>>

Class ImplicitGeometry	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«ObjectType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	ImplicitGeometry
Target Role:	appearance
Target Class:	AbstractAppearance
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	ImplicitGeometry
Target Role:	relativeGMLGeometry
Target Class:	GM_Object
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	ImplicitGeometry
Target Role:	referencePoint
Target Class:	GM_Point

Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractOccupiedSpace Target Role: lod2ImplicitRepresentation Target Class: ImplicitGeometry
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractOccupiedSpace Target Role: lod1ImplicitRepresentation Target Class: ImplicitGeometry
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractOccupiedSpace Target Role: lod3ImplicitRepresentation Target Class: ImplicitGeometry
Attributes
Attribute Name: libraryObject Value Type: URI Definition: SIG3D: External link to a prototype geometry. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: mimeType Value Type: MimeValue Definition: SIG3D: Mime type of the referenced external geometric object (attribute libraryObject). Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: transformationMatrix Value Type: TransformationMatrix4x4 Definition: SIG3D: Mathematical transformation (translation, rotation and scaling) between the prototype geometry and the actual spatial position of the object. Multiplicity: Stereotype: «Property»

12.2.23. Class IntegerBetween0and3

Subclass of <←section,>>

Class IntegerBetween0and3
Definition: Subtype Of: <←section,>> StereoType: «BasicType»
Associations
Name: Type: NoteLink Direction: Source Role: Source Class: Note Target Role: Target Class: IntegerBetween0and3
Attributes

12.2.24. Class IntervalValue

Subclass of <←section,>>

Class IntervalValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: IntervalValue
Attributes

12.2.25. Class MimeValue

Subclass of <←section,>>

Class MimeValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations

Attributes

12.2.26. Class OccupantTypeValue

Subclass of <←section,>>

Class OccupantTypeValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»

Associations
Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: OccupantTypeValue
Attributes

12.2.27. Class OtherRelationTypeValue

Subclass of RelationTypeValue

Class OtherRelationTypeValue
Definition: Subtype Of: RelationTypeValue StereoType: «CodeList»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: OtherRelationTypeValue Target Role: Target Class: RelationTypeValue
Attributes

12.2.28. Class QualifiedAreaValue

Subclass of <←section,>>

Class QualifiedAreaValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»

Associations	
Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	QualifiedAreaValue

Attributes

12.2.29. Class QualifiedVolumeValue

Subclass of <←section,>>

Class QualifiedVolumeValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»

Associations	
Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	QualifiedVolumeValue

Attributes

12.2.30. Class RelationTypeValue

Subclass of <←section,>>

Class RelationTypeValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»

Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TemporalRelationTypeValue Target Role: Target Class: RelationTypeValue
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TopologicRelationTypeValue Target Role: Target Class: RelationTypeValue
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: OtherRelationTypeValue Target Role: Target Class: RelationTypeValue
Attributes

12.2.31. Class TemporalRelationTypeValue

Subclass of RelationTypeValue

Class TemporalRelationTypeValue
Definition: Subtype Of: RelationTypeValue StereoType: «CodeList»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TemporalRelationTypeValue Target Role: Target Class: RelationTypeValue
Attributes

12.2.32. Class TopologicRelationTypeValue

Subclass of RelationTypeValue

Class TopologicRelationTypeValue
Definition: Subtype Of: RelationTypeValue StereoType: «CodeList»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TopologicRelationTypeValue Target Role: Target Class: RelationTypeValue
Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: TopologicRelationTypeValue
Attributes

12.2.33. Class TransformationMatrix2x2

Subclass of DoubleList

Class TransformationMatrix2x2
Definition: Subtype Of: DoubleList StereoType: «BasicType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TransformationMatrix2x2 Target Role: Target Class: DoubleList

Name: Type: NoteLink Direction: Source Role: Source Class: Note Target Role: Target Class: TransformationMatrix2x2
Attributes

12.2.34. Class TransformationMatrix3x4

Subclass of DoubleList

Class TransformationMatrix3x4
Definition: Subtype Of: DoubleList StereoType: «BasicType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TransformationMatrix3x4 Target Role: Target Class: DoubleList
Name: Type: NoteLink Direction: Source Role: Source Class: Note Target Role: Target Class: TransformationMatrix3x4
Attributes

12.2.35. Class TransformationMatrix4x4

Subclass of DoubleList

Class TransformationMatrix4x4
Definition: Subtype Of: DoubleList StereoType: «BasicType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TransformationMatrix4x4
Target Role:	
Target Class:	DoubleList
Name:	
Type:	NoteLink
Direction:	
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	TransformationMatrix4x4
Attributes	

12.2.36. Class AbstractGenericAttribute

Subclass of <←section,>>

Class AbstractGenericAttribute	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«DataType»

Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CityObjectRelation
Target Role:	genericAttribute
Target Class:	AbstractGenericAttribute
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GenericAttributeSet
Target Role:	
Target Class:	AbstractGenericAttribute

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	StringAttribute
Target Role:	
Target Class:	AbstractGenericAttribute
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	UriAttribute
Target Role:	
Target Class:	AbstractGenericAttribute
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	GenericAttributeSet
Target Role:	genericAttribute
Target Class:	AbstractGenericAttribute
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	MeasureAttribute
Target Role:	
Target Class:	AbstractGenericAttribute
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	DoubleAttribute
Target Role:	
Target Class:	AbstractGenericAttribute
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	IntAttribute
Target Role:	
Target Class:	AbstractGenericAttribute

Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractCityObject Target Role: genericAttribute Target Class: AbstractGenericAttribute
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: DateAttribute Target Role: Target Class: AbstractGenericAttribute
Attributes

12.2.37. Class ExternalReference

Subclass of <←section,>>

Class ExternalReference
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractCityObject Target Role: externalReference Target Class: ExternalReference
Attributes
Attribute Name: informationSystem Value Type: URI Definition: SIG3D: URL or URN of the information system or data set. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: relationType Value Type: URI Definition: Multiplicity: [0..1] Stereotype: «Property»

Attribute Name: targetResource Value Type: URI Definition: SIG3D: Referenced object in the external information system or data set. Multiplicity: Stereotype: «Property»

12.2.38. Class Occupancy

Subclass of <←section,>>

Class Occupancy
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Attributes
Attribute Name: interval Value Type: IntervalValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: numberOfOccupants Value Type: Integer Definition: Multiplicity: Stereotype: «Property»
Attribute Name: occupantType Value Type: OccupantTypeValue Definition: Multiplicity: [0..1] Stereotype: «Property»

12.2.39. Class QualifiedArea

Subclass of <←section,>>

Class QualifiedArea
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Attributes

Attribute Name: area Value Type: Area Definition: Multiplicity: Stereotype: «Property»
Attribute Name: typeOfArea Value Type: QualifiedAreaValue Definition: Multiplicity: Stereotype: «Property»

12.2.40. Class QualifiedVolume

Subclass of <←section,>>

Class QualifiedVolume
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Attributes
Attribute Name: typeOfVolume Value Type: QualifiedVolumeValue Definition: Multiplicity: Stereotype: «Property»
Attribute Name: volume Value Type: Volume Definition: Multiplicity: Stereotype: «Property»

12.2.41. Class RelativeToTerrain

Subclass of <←section,>>

Class RelativeToTerrain
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes

Attribute Name: entirelyAboveTerrain Value Type: Definition: SIG3D: Object is located entirely above terrain. Multiplicity: Stereotype:
Attribute Name: substantiallyAboveTerrain Value Type: Definition: SIG3D: Most, but not all parts of the object are located above terrain. Multiplicity: Stereotype:
Attribute Name: substantiallyAboveAndBelowTerrain Value Type: Definition: SIG3D: Parts of the object are located above terrain, and other parts below terrain. Multiplicity: Stereotype:
Attribute Name: substantiallyBelowTerrain Value Type: Definition: SIG3D: Most, but not all parts of the object are located below terrain. Multiplicity: Stereotype:
Attribute Name: entirelyBelowTerrain Value Type: Definition: SIG3D: All parts of the object are located below terrain. Multiplicity: Stereotype:

12.2.42. Class RelativeToWater

Subclass of <←section,>>

Class RelativeToWater
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes
Attribute Name: entirelyAboveWaterSurface Value Type: Definition: SIG3D: Object is located entirely above water surface. Multiplicity: Stereotype:

Attribute Name:	substantiallyAboveWaterSurface
Value Type:	
Definition:	SIG3D: Most, but not all parts of the object are located above water surface.
Multiplicity:	
Stereotype:	
Attribute Name:	substantiallyAboveAndBelowWaterSurface
Value Type:	
Definition:	SIG3D: Parts of the object are located above water surface, and other parts below water surface.
Multiplicity:	
Stereotype:	
Attribute Name:	substantiallyBelowWaterSurface
Value Type:	
Definition:	SIG3D: Most, but not all parts of the object are located below water surface.
Multiplicity:	
Stereotype:	
Attribute Name:	entirelyBelowWaterSurface
Value Type:	
Definition:	SIG3D: All parts of the object are located below water surface.
Multiplicity:	
Stereotype:	
Attribute Name:	temporarilyAboveAndBelowWaterSurface
Value Type:	
Definition:	SIG3D: The height of the water surface is varying and the object temporily is located above or below water level.
Multiplicity:	
Stereotype:	

12.2.43. Class SpaceType

Subclass of <←section,>>

Class SpaceType	
Definition:	
Subtype Of:	<←section,>>
StereoType:	
Associations	
Attributes	
Attribute Name:	closed
Value Type:	
Definition:	boundaries at all sides
Multiplicity:	
Stereotype:	

Attribute Name:	open
Value Type:	
Definition:	boundary at the bottom only
Multiplicity:	
Stereotype:	
Attribute Name:	semiOpen
Value Type:	
Definition:	at least one side has a boundary
Multiplicity:	
Stereotype:	

12.2.44. Class XALAddressDetails

Subclass of <←section,>>

Class XALAddressDetails	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«DataType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Address
Target Role:	xalAddress
Target Class:	XALAddressDetails
Attributes	

12.2.45. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.2.46. Requirements

Requirement 5	**/req/core/base**

A	The CityModel element (type: CityModelType, substitutionGroup: gml:_FeatureCollection) shall only contain cityObjectMember elements (type: gml:FeaturePropertyType), app:appearanceMember ele-ments (type: app:AppearancePropertyType), and gml:featureMember elements (type: gml:FeaturePropertyType) as feature members.
B	The type ExternalObjectReference introduces the two elements name (type: xs:string) and uri (type: xs:anyURI). The external reference may be specified by either of them. However, if the informationSys-tem property element (type: xs:anyURI) of the type ExternalReferenceType is not provided, the uri ele-ment of ExternalObjectReference must be given.
C	In order to represent address information about a feature, the corresponding thematic class of the feature shall define a property of the type AddressPropertyType. Thus, for all CityGML extension modules on-ly the type AddressPropertyType shall be used for elements providing address information.
D	Since the concept of implicit geometries (cf. chapter 8.2) is part of the CityGML Core module, the con-formance requirements introduced for implicit geometries (cf. chapter 8.3.3) are part of the conform-ance requirements of the core.

Requirement 6	**/req/core/refIntegrity**
A	The cityObjectMember element (type: gml:FeaturePropertyType) may contain a _CityObject element, which typically is an object from a derived subclass like bldg:Building, inline or an XLink reference to a remote _CityObject element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the cityObjectMember element may only point to a remote _CityObject element (where re-mote _CityObject elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
B	The type AddressPropertyType may contain an Address element inline or an XLink reference to a re-mote Address element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the corresponding element of type AddressPropertyType may only point to a remote Address element (where remote Address elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

12.3. Digital Terrain Model

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-relief	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

An essential part of a city model is the terrain. The Digital Terrain Model (DTM) of CityGML is provided by the thematic extension module Relief (cf. chapter 7). In CityGML, the terrain is represented by the class ReliefFea-ture in LOD 0-4 (Fig. 24 depicts the UML diagram, for the XML schema definition see annex A.9). A Re-liefFeature consists of one or more entities of the class ReliefComponent. Its validity may be restricted to a certain area defined by an optional validity extent polygon. As ReliefFeature and ReliefComponent are deriva-tives of _CityObject, the corresponding attributes and relations are inherited. The class ReliefFeature is associat-ed with different concepts of terrain representations which can coexist. The terrain may be specified as a regular raster or grid (RasterRelief), as a TIN (Triangulated Irregular Network, TINReflief), by break lines (BreaklineRe-lief), or by mass points (MasspointRelief). The four types are implemented by the corresponding GML3 classes: grids by gml:RectifiedGridCoverage, break lines by gml:MultiCurve, mass points by gml:MultiPoint and TINs either by gml:TriangulatedSurface or by gml:Tin. In case of gml:TriangulatedSurfaces, the triangles are given explicitly while in case of gml:Tin only 3D points are represented, where the triangulation can be reconstructed by standard methods (Delaunay triangulation, cf. Okabe et al. 1992). Break lines are represented by 3D curves. Mass points are simply a set of 3D points.

The UML diagram of the Relief Package is depicted in Relief UML Diagram. The Data Dictionary for the Relief Package is provided in section Relief Data Dictionary.

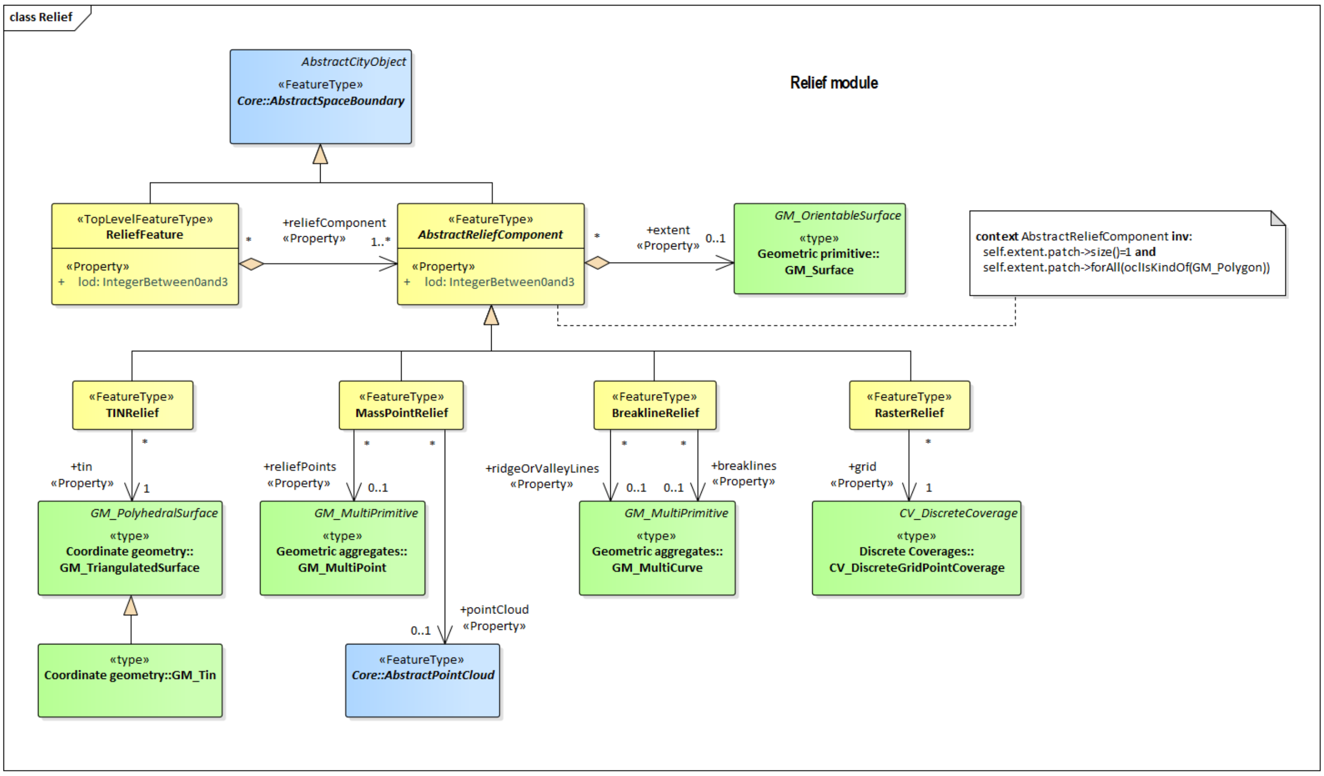


Figure 16. UML diagram of Relief module.

The UML diagram of Relief module. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.4. Relief Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.4.1. Class AbstractReliefComponent

Subclass of AbstractSpaceBoundary

Class AbstractReliefComponent
Definition: Subtype Of: AbstractSpaceBoundary StereoType: «FeatureType»
Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractReliefComponent
Target Role:	
Target Class:	AbstractSpaceBoundary

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractReliefComponent
Target Role:	extent
Target Class:	GM_Surface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BreaklineRelief
Target Role:	
Target Class:	AbstractReliefComponent

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TINRelief
Target Role:	
Target Class:	AbstractReliefComponent

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	MassPointRelief
Target Role:	
Target Class:	AbstractReliefComponent

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	ReliefFeature
Target Role:	reliefComponent
Target Class:	AbstractReliefComponent

Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: AbstractReliefComponent
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: RasterRelief Target Role: Target Class: AbstractReliefComponent
Attributes
Attribute Name: lod Value Type: IntegerBetween0and3 Definition: SIG3D: Number denoting the LOD of the relief component. The LOD concept for the relief is defined in chapter … Multiplicity: Stereotype: «Property»

12.4.2. Class BreaklineRelief

Subclass of AbstractReliefComponent

Class BreaklineRelief
Definition: Subtype Of: AbstractReliefComponent StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: BreaklineRelief Target Role: ridgeOrValleyLines Target Class: GM_MultiCurve

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BreaklineRelief Target Role: Target Class: AbstractReliefComponent
Name: Type: Association Direction: Source → Destination Source Role: Source Class: BreaklineRelief Target Role: breaklines Target Class: GM_MultiCurve
Attributes

12.4.3. Class MassPointRelief

Subclass of AbstractReliefComponent

Class MassPointRelief
Definition: Subtype Of: AbstractReliefComponent StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: MassPointRelief Target Role: reliefPoints Target Class: GM_MultiPoint
Name: Type: Association Direction: Source → Destination Source Role: Source Class: MassPointRelief Target Role: pointCloud Target Class: AbstractPointCloud

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: MassPointRelief Target Role: Target Class: AbstractReliefComponent
Attributes

12.4.4. Class RasterRelief

Subclass of AbstractReliefComponent

Class RasterRelief
Definition: Subtype Of: AbstractReliefComponent StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: RasterRelief Target Role: Target Class: AbstractReliefComponent
Name: Type: Association Direction: Source → Destination Source Role: Source Class: RasterRelief Target Role: grid Target Class: CV_DiscreteGridPointCoverage
Attributes

12.4.5. Class ReliefFeature

Subclass of AbstractSpaceBoundary

Class ReliefFeature
Definition: Subtype Of: AbstractSpaceBoundary StereoType: «TopLevelFeatureType»

Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: ReliefFeature Target Role: reliefComponent Target Class: AbstractReliefComponent
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: ReliefFeature Target Role: Target Class: AbstractSpaceBoundary
Attributes
Attribute Name: lod Value Type: IntegerBetween0and3 Definition: SIG3D: Number denoting the LOD of the relief feature. The LOD concept for the relief is defined in chapter ... Multiplicity: Stereotype: «Property»

12.4.6. Class TINRelief

Subclass of AbstractReliefComponent

Class TINRelief
Definition: Subtype Of: AbstractReliefComponent StereoType: «FeatureType»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TINRelief Target Role: Target Class: AbstractReliefComponent

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	TINRelief
Target Role:	tin
Target Class:	GM_TriangulatedSurface
Attributes	

12.4.7. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.4.8. Requirements

Requirement 7	**/req/relief/base**
A	The gml:Polygon geometry element describing the extent of validity of a _ReliefComponent element us-ing the extent property (type: gml:PolygonPropertyType) of _ReliefComponent shall be given as 2D footprint polygon which may have inner holes.

Requirement 8	**/req/relief/refIntegrity**
A	The reliefComponent property (type: ReliefComponentPropertyType) of the element ReliefFeature may contain a _ReliefComponent element inline or an XLink reference to a remote _ReliefComponent ele-ment using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the reliefCom-ponent property may only point to a remote _ReliefComponent element (where remote _ReliefComponent elements are located in another document or elsewhere in the same document). Ei-ther the contained element or the reference must be given, but neither both nor none.

B	The tin property (type: tinPropertyType) of the element TINRelief may contain a gml:TriangulatedSurface element inline or an XLink reference to a remote gml:TriangulatedSurface el-ement using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the tin prop-erty may only point to a remote gml:TriangulatedSurface element (where remote gml:TriangulatedSurface elements are located in another document or elsewhere in the same docu-ment). Either the contained element or the reference must be given, but neither both nor none.
C	The grid property (type: gridPropertyType) of the element RasterRelief may contain a gml:RectifiedGridCoverage element inline or an XLink reference to a remote gml:RectifiedGridCoverage element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the grid property may only point to a remote gml:RectifiedGridCoverage element (where remote gml:RectifiedGridCoverage elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

12.5. Building Model

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-building	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

The building model is one of the most detailed thematic concepts of CityGML. It allows for the representation of thematic and spatial aspects of buildings and building parts in five levels of detail, LOD0 to LOD4. The building model of CityGML is defined by the thematic extension module Building (cf. chapter 7). Fig. 26 provides examples of 3D city and building models in LOD1 – 4.

The UML diagram of the building model is depicted in Building UML Diagram. The Data Dictionary for the Building Model Package is provided in section Building Model Data Dictionary.

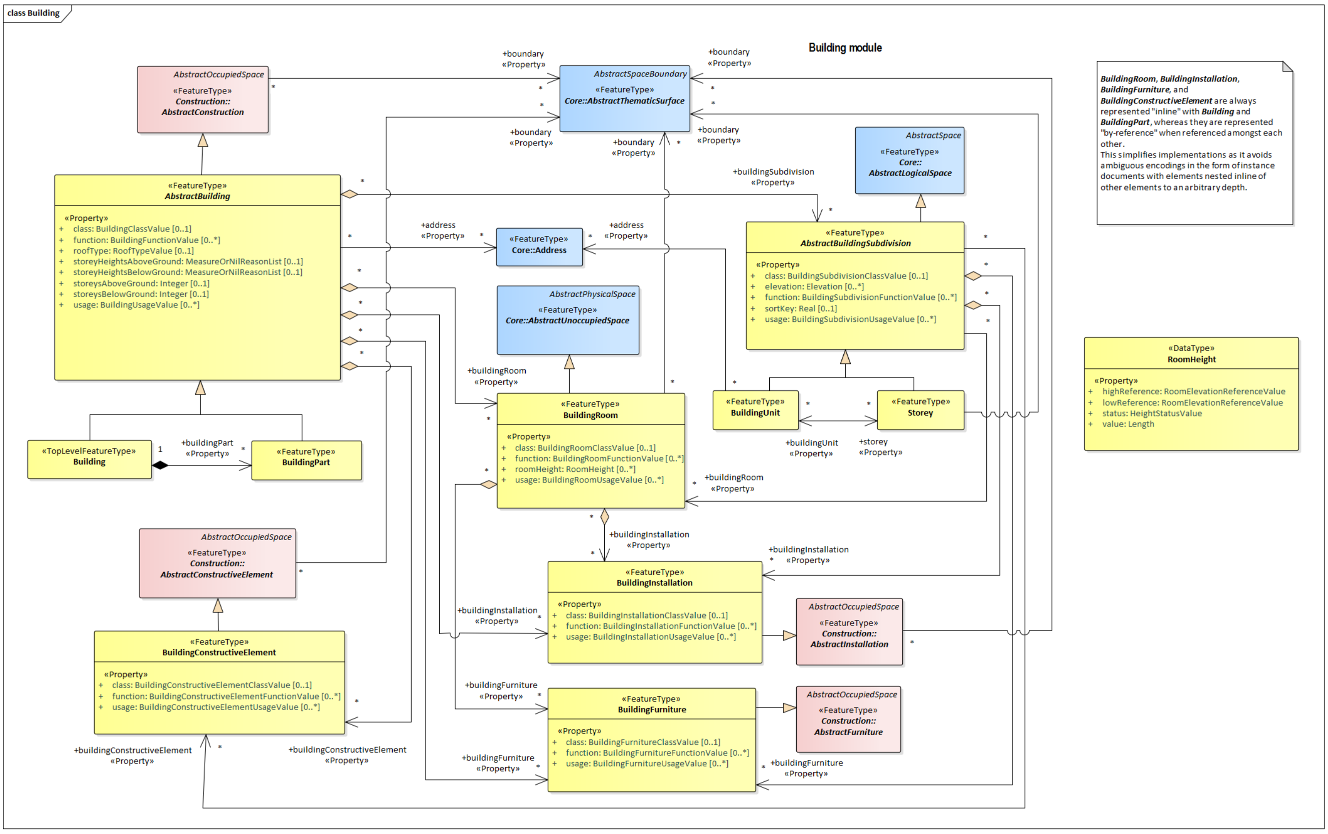


Figure 17. UML diagram of CityGML's building model.

The UML diagram of CityGML's building model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.6. Building Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.6.1. Class AbstractBuilding

Subclass of AbstractConstruction

Class AbstractBuilding
Definition: Subtype Of: AbstractConstruction StereoType: «FeatureType»
Associations

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingConstructiveElement
Target Class:	BuildingConstructiveElement

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	address
Target Class:	Address

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingSubdivision
Target Class:	AbstractBuildingSubdivision

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	
Target Class:	AbstractConstruction

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingFurniture
Target Class:	BuildingFurniture

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingRoom
Target Class:	BuildingRoom

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: AbstractBuilding
Target Role: buildingInstallation
Target Class: BuildingInstallation

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: BuildingPart
Target Role:
Target Class: AbstractBuilding

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: Building
Target Role:
Target Class: AbstractBuilding

Attributes

Attribute Name: class
Value Type: BuildingClassValue
Definition: SIG3D: Classification of Building or BuildingPart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: BuildingFunctionValue
Definition: SIG3D: Specified function of Building or BuildingPart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: roofType
Value Type: RoofTypeValue
Definition: bSI: Basic configuration of the roof in terms of the different roof shapes.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: storeyHeightsAboveGround
Value Type: MeasureOrNilReasonList
Definition: SIG3D: List of heights for each storey above ground.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name:	storeyHeightsBelowGround
Value Type:	MeasureOrNilReasonList
Definition:	SIG3D: List of heights for each storey below ground.
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	storeysAboveGround
Value Type:	Integer
Definition:	SIG3D: Number of storeys mainly above ground.
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	storeysBelowGround
Value Type:	Integer
Definition:	SIG3D: Number of storeys mainly below ground.
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	usage
Value Type:	BuildingUsageValue
Definition:	SIG3D: Actual usage of Building or BuildingPart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity:	[0..*]
Stereotype:	«Property»

12.6.2. Class AbstractBuildingSubdivision

Subclass of AbstractLogicalSpace

Class AbstractBuildingSubdivision	
Definition:	
Subtype Of:	AbstractLogicalSpace
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	
Target Class:	AbstractLogicalSpace

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingInstallation
Target Class:	BuildingInstallation

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingFurniture
Target Class:	BuildingFurniture

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingRoom
Target Class:	BuildingRoom

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingConstructiveElement
Target Class:	BuildingConstructiveElement

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Storey
Target Role:	
Target Class:	AbstractBuildingSubdivision

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingSubdivision
Target Class:	AbstractBuildingSubdivision

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BuildingUnit Target Role: Target Class: AbstractBuildingSubdivision
Attributes
Attribute Name: class Value Type: BuildingSubdivisionClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: elevation Value Type: Elevation Definition: [INSPIRE] Vertically-constrained dimensional property consisting of an absolute measure referenced to a well-defined surface which is commonly taken as origin (geoïd, water level, etc.). Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: function Value Type: BuildingSubdivisionFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: sortKey Value Type: Real Definition: Defines an order among the building unit objects, e.g. for sorting storeys. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: usage Value Type: BuildingSubdivisionUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.6.3. Class Building

Subclass of AbstractBuilding

Class Building
Definition: Subtype Of: AbstractBuilding StereoType: «TopLevelFeatureType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: Building
Target Role:
Target Class: AbstractBuilding

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: Building
Target Role: buildingPart
Target Class: BuildingPart

Attributes

12.6.4. Class BuildingClassValue

Subclass of <←section,>>

Class BuildingClassValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.6.5. Class BuildingConstructiveElement

Subclass of AbstractConstructiveElement

Class BuildingConstructiveElement

Definition:
Subtype Of: AbstractConstructiveElement
StereoType: «FeatureType»

Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingConstructiveElement
Target Role:	
Target Class:	AbstractConstructiveElement
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingConstructiveElement
Target Class:	BuildingConstructiveElement
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingConstructiveElement
Target Class:	BuildingConstructiveElement
Attributes	
Attribute Name:	class
Value Type:	BuildingConstructiveElementClassValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»
Attribute Name:	function
Value Type:	BuildingConstructiveElementFunctionValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»
Attribute Name:	usage
Value Type:	BuildingConstructiveElementUsageValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

12.6.6. Class BuildingConstructiveElementClassValue

Subclass of <←section,>>

Class BuildingConstructiveElementClassValue

Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.7. Class BuildingConstructiveElementFunctionValue

Subclass of <←section,>>

Class BuildingConstructiveElementFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.8. Class BuildingConstructiveElementUsageValue

Subclass of <←section,>>

Class BuildingConstructiveElementUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.9. Class BuildingFunctionValue

Subclass of <←section,>>

Class BuildingFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.10. Class BuildingFurniture

Subclass of AbstractFurniture

Class BuildingFurniture
Definition: Subtype Of: AbstractFurniture StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BuildingFurniture Target Role: Target Class: AbstractFurniture
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBuildingSubdivision Target Role: buildingFurniture Target Class: BuildingFurniture
Name: Type: Association Direction: Source → Destination Source Role: Source Class: BuildingRoom Target Role: buildingFurniture Target Class: BuildingFurniture
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBuilding Target Role: buildingFurniture Target Class: BuildingFurniture
Attributes
Attribute Name: class Value Type: BuildingFurnitureClassValue Definition: SIG3D: Classification of BuildingFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»

Attribute Name: function Value Type: BuildingFurnitureFunctionValue Definition: SIG3D: Specified function of BuildingFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BuildingFurnitureUsageValue Definition: SIG3D: Actual usage of BuildingFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.6.11. Class BuildingFurnitureClassValue

Subclass of <←section,>>

Class BuildingFurnitureClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.12. Class BuildingFurnitureFunctionValue

Subclass of <←section,>>

Class BuildingFurnitureFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.13. Class BuildingFurnitureUsageValue

Subclass of <←section,>>

Class BuildingFurnitureUsageValue

Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»
Associations	
Attributes	

12.6.14. Class BuildingInstallation

Subclass of AbstractInstallation

Class BuildingInstallation	
Definition:	
Subtype Of:	AbstractInstallation
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingInstallation
Target Role:	
Target Class:	AbstractInstallation
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingInstallation
Target Class:	BuildingInstallation
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	buildingInstallation
Target Class:	BuildingInstallation
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	buildingInstallation
Target Class:	BuildingInstallation

Attributes
Attribute Name: class Value Type: BuildingInstallationClassValue Definition: SIG3D: Classification of BuildingInstallation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: BuildingInstallationFunctionValue Definition: SIG3D: Specified function of BuildingInstallation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BuildingInstallationUsageValue Definition: SIG3D: Actual usage of BuildingInstallation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.6.15. Class BuildingInstallationClassValue

Subclass of <←section,>>

Class BuildingInstallationClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.16. Class BuildingInstallationFunctionValue

Subclass of <←section,>>

Class BuildingInstallationFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.17. Class BuildingInstallationUsageValue

Subclass of <←section,>>

Class BuildingInstallationUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.18. Class BuildingPart

Subclass of AbstractBuilding

Class BuildingPart
Definition: Subtype Of: AbstractBuilding StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BuildingPart Target Role: Target Class: AbstractBuilding
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Building Target Role: buildingPart Target Class: BuildingPart
Attributes

12.6.19. Class BuildingRoom

Subclass of AbstractUnoccupiedSpace

Class BuildingRoom

Definition:	
Subtype Of:	AbstractUnoccupiedSpace
StereoType:	«FeatureType»

Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	
Target Class:	AbstractUnoccupiedSpace

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	buildingFurniture
Target Class:	BuildingFurniture

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingRoom
Target Role:	buildingInstallation
Target Class:	BuildingInstallation

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuildingSubdivision
Target Role:	buildingRoom
Target Class:	BuildingRoom

Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBuilding Target Role: buildingRoom Target Class: BuildingRoom
Attributes
Attribute Name: class Value Type: BuildingRoomClassValue Definition: SIG3D: Classification of Room as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: BuildingRoomFunctionValue Definition: SIG3D: Function of Room as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: roomHeight Value Type: RoomHeight Definition: Height of the room. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BuildingRoomUsageValue Definition: SIG3D: Usage of Room as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.6.20. Class BuildingRoomClassValue

Subclass of <←section,>>

Class BuildingRoomClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.21. Class BuildingRoomFunctionValue

Subclass of <←section,>>

Class BuildingRoomFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.22. Class BuildingRoomUsageValue

Subclass of <←section,>>

Class BuildingRoomUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.23. Class BuildingSubdivisionClassValue

Subclass of <←section,>>

Class BuildingSubdivisionClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.24. Class BuildingSubdivisionFunctionValue

Subclass of <←section,>>

Class BuildingSubdivisionFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»

Associations
Attributes

12.6.25. Class BuildingSubdivisionUsageValue

Subclass of <←section,>>

Class BuildingSubdivisionUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.26. Class BuildingUnit

Subclass of AbstractBuildingSubdivision

Class BuildingUnit
Definition: Subtype Of: AbstractBuildingSubdivision StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: BuildingUnit Target Role: address Target Class: Address
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BuildingUnit Target Role: Target Class: AbstractBuildingSubdivision

Name: Type: Association Direction: Bi-Directional Source Role: storey Source Class: Storey Target Role: buildingUnit Target Class: BuildingUnit
Attributes

12.6.27. Class BuildingUsageValue

Subclass of <←section,>>

Class BuildingUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.28. Class RoofTypeValue

Subclass of <←section,>>

Class RoofTypeValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.6.29. Class RoomElevationReferenceValue

Subclass of <←section,>>

Class RoomElevationReferenceValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations

Attributes

12.6.30. Class Storey

Subclass of AbstractBuildingSubdivision

Class Storey
Definition: Subtype Of: AbstractBuildingSubdivision StereoType: «FeatureType»

Associations
Name: Type: Association Direction: Bi-Directional Source Role: storey Source Class: Storey Target Role: buildingUnit Target Class: BuildingUnit
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Storey Target Role: Target Class: AbstractBuildingSubdivision
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Storey Target Role: boundary Target Class: AbstractThematicSurface
Attributes

12.6.31. Class RoomHeight

Subclass of <←section,>>

Class RoomHeight
Definition: Subtype Of: <←section,>> StereoType: «DataType»

Associations
Attributes
Attribute Name: highReference Value Type: RoomElevationReferenceValue Definition: [INSPIRE] Element used as the high reference. Multiplicity: Stereotype: «Property»
Attribute Name: lowReference Value Type: RoomElevationReferenceValue Definition: [INSPIRE] Element as the low reference. Multiplicity: Stereotype: «Property»
Attribute Name: status Value Type: HeightStatusValue Definition: [INSPIRE] The way the height has been captured. Multiplicity: Stereotype: «Property»
Attribute Name: value Value Type: Length Definition: Value of the room height. Multiplicity: Stereotype: «Property»

12.6.32. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

NOTE	The following content is from the previous draft. Is this sort of content useful?

BuildingType, Building

The Building class is one of the two subclasses of _AbstractBuilding. If a building only consists of one (homo-geneous) part, this class shall be used. A building composed of structural segments differing in, for example the number of storeys or the roof type has to be separated into one Building having one or more additional Build-ingPart (see Fig. 28). The geometry and non-spatial properties of the central part of the building should be represented in the aggregating Building feature.

BuildingType, Building Part

The class BuildingPart is derived from _AbstractBuilding. It is used to model a structural part of a building (see Fig. 28). A BuildingPart object should be uniquely related to exactly one building or building part object.

AbstractBuildingType, _AbstractBuilding

The abstract class _AbstractBuilding contains properties for building attributes, purely geometric representations, and geometric/semantic representations of the building or building part in different levels of detail. The attributes describe:

1. classification of the building or building part (class), the different intended usages (function), and the different actual usages (usage). The permitted values for these attributes can be specified in code lists.
2. The year of construction (yearOfConstruction) and the year of demolition (yearOfDemolition) of the build-ing or building part. These attributes can be used to describe the chronology of the building development within a city model. The points of time refer to real world time.
3. The roof type of the building or building part (roofType). The permitted values for this attribute can be specified in a code list.
4. The measured relative height (measuredHeight) of the building or building part.
5. The number of storeys above (storeyAboveGround) and below (storeyBelowGround) ground level.
6. The list of storey heights above (storeyHeightsAboveGround) and below (storeyHeightsBelowGround) ground level. The first value in a list denotes the height of the nearest storey wrt. to the ground level and last value the height of the farthest.

Table 6. Semantic themes of the class _AbstractBuilding

Geometric / semantic theme	Property type	LOD0	LOD1	LOD2	LOD3	LOD4
Building footprint and roof edge	gml:MultiSurfaceType	•				
Volume part of the building shell	gml:SolidType		•	•	•	•
Surface part of the building shell	gml:MultiSurfaceType		•	•	•	•
Terrain intersection curve	gml:MultiCurveType		•	•	•	•
Curve part of the building shell	gml:MultiCurveType			•	•	•
Building parts	BuildingPartType		•	•	•	•
Boundary surfaces (chapter 10.3.3)	AbstractBoundarySurfaceType			•	•	•
Outer building installations (chapter 10.3.2)	BuildingInstallationType			•	•	•

Geometric / semantic theme	Property type	LOD0	LOD1	LOD2	LOD3	LOD4
Openings (chapter 10.3.4)	AbstractOpeningType				•	•
Rooms (chapter 10.3.5)	RoomType					•
Interior building installations (chapter 10.3.5)	IntBuildingInstallationType					•

BuildingInstallationType, BuildingInstallation

Note: insert BuildingInstallation UML

12.6.33. Boundary surfaces

AbstractBoundarySurfaceType, _BoundarySurface

NOTE Insert AbstractBoundarySurfaceType, _BoundarySurface UML

GroundSurfaceType, GroundSurface

NOTE insert GroundSurfaceType, GroundSurface uml

The ground plate of a building or building part is modelled by the class GroundSurface. The polygon defining the ground plate is congruent with the building's footprint. However, the surface normal of the ground plate is pointing downwards.

OuterCeilingSurfaceType, OuterCeilingSurface

NOTE insert OuterCeilingSurfaceType, OuterCeilingSurface UML

A mostly horizontal surface belonging to the outer building shell and having the orientation pointing downwards can be modeled as an OuterCeilingSurface. Examples are the visible part of the ceiling of a loggia or the ceiling of a passage.

WallSurfaceType, WallSurface

NOTE insert WallSurfaceType, WallSurface UML

All parts of the building facade belonging to the outer building shell can be modelled by the class WallSurface.

OuterFloorSurfaceType, OuterFloorSurface

NOTE insert OuterFloorSurfaceType, OuterFloorSurface UML

A mostly horizontal surface belonging to the outer building shell and with the orientation pointing upwards can be modeled as an OuterFloorSurface. An example is the floor of a loggia.

RoofSurfaceType, RoofSurface

NOTE insert RoofSurfaceType, RoofSurface UML

The major roof parts of a building or building part are expressed by the class RoofSurface. Secondary parts of a roof with a specific semantic meaning like dormers or chimneys should be modelled as BuildingInstallation.

ClosureSurfaceType, ClosureSurface

NOTE insert ClosureSurfaceType, ClosureSurface UML

An opening in a building not filled by a door or window can be sealed by a virtual surface called ClosureSurface (cf. chapter 6.4). Hence, buildings with open sides like a barn or a hangar, can be virtually closed in order to be able to compute their volume. ClosureSurfaces are also used in the interior building model. If two rooms with a different function (e.g. kitchen and living room) are directly connected without a separating door, a ClosureSur-face should be used to separate or connect the volumes of both rooms.

FloorSurfaceType, FloorSurface

NOTE insert FloorSurfaceType, FloorSurface UML

The class FloorSurface must only be used in the LOD4 interior building model for modelling the floor of a room.

InteriorWallSurfaceType, InteriorWallSurface

NOTE insert InteriorWallSurfaceType, InteriorWallSurface UML

The class InteriorWallSurface must only be used in the LOD4 interior building model for modelling the visible surfaces of the room walls.

CeilingSurfaceType, CeilingSurface

NOTE Insert CeilingSurfaceType, CeilingSurface UML

The class CeilingSurface must only be used in the LOD4 interior building model for modelling the ceiling of a room.

12.6.34. Openings

AbstractOpeningType, _Opening

NOTE insert AbstractOpeningType, _Opening UML

The class _Opening is the abstract base class for semantically describing openings like doors or windows in outer or inner boundary surfaces like walls and roofs. Openings only exist in models of LOD3 or LOD4. Each _Opening is associated with a gml:MultiSurface geometry. Alternatively, the geometry may be given as Implic-itGeometry object. Following the concept of ImplicitGeometry the geometry of a prototype opening is stored only once in a local coordinate system and referenced by other opening features (see chapter 8.2).

WindowType, Window

NOTE	insert WindowType, Window UML

The class Window is used for modelling windows in the exterior shell of a building, or hatches between adjacent rooms. The formal difference between the classes Window and Door is that – in normal cases – Windows are not specifically intended for the transit of people or vehicles.

DoorType, Door

NOTE	insert DoorType, Door UML

The class Door is used for modelling doors in the exterior shell of a building, or between adjacent rooms. Doors can be used by people to enter or leave a building or room. In contrast to a ClosureSurface a door may be closed, blocking the transit of people. A Door may be assigned zero or more addresses. The corresponding Address-PropertyType is defined within the CityGML core module (cf. chapter 10.1.4) .

12.6.35. Building Interior

RoomType, Room

NOTE	insert RoomType, Room UML

A Room is a semantic object for modelling the free space inside a building and should be uniquely related to exactly one building or building part object. It should be closed (if necessary by using ClosureSurfaces) and the geometry normally will be described by a solid (lod4Solid). However, if the topological correctness of the boundary cannot be guaranteed, the geometry can alternatively be given as a MultiSurface (lod4MultiSurface). The surface normals of the outer shell of a GML solid must point outwards. This is important to consider when Room surfaces should be assigned Appearances. In this case, textures and colors must be placed on the backside of the corresponding surfaces in order to be visible from the inside of the room.

BuildingFurnitureType, BuildingFurniture

NOTE	insert BuildingFurnitureType, BuildingFurniture UML

Rooms may have BuildingFurnitures and IntBuildingInstallations. A BuildingFurniture is a movable part of a room, such as a chair or furniture. A BuildingFurniture object should be uniquely related to exactly one room object. Its geometry may be represented by an explicit geometry or an ImplicitGeometry object. Following the concept of ImplicitGeometry the geometry of a prototype building furniture is stored only once in a local coordi-nate system and referenced by other building furniture features (see chapter 8.2).

IntBuildingInstallationType, IntBuildingInstallation

NOTE insert IntBuildingInstallationType, IntBuildingInstallation UML

12.6.36. Modelling building storeys using CityObjectGroups

CityGML does currently not provide a specific concept for the representation of storeys as it is available in the AEC/FM standard IFC (IAI 2006). However, a storey can be represented as an explicit aggregation of all build-ing features on a certain height level using CityGML's notion of CityObjectGroups (cf. chapter 10.11).

In order to model building storeys with CityGML's generic grouping concept, a nested hierarchy of CityObject-Group objects has to be used. In a first step, all semantic objects belonging to a specific storey are grouped. The attributes of the corresponding CityObjectGroup object are set as follows:

- The class attribute shall be assigned the value “building separation”.
- The function attribute shall be assigned the value “lodXStorey” with X between 1 and 4 in order to de-note that this group represents a storey wrt. a specific LOD.
- The storey name or number can be stored in the gml:name property. The storey number attribute shall be assigned the value “storeyNo_X” with decimal number X in order to denote that this group repre-sents a storey wrt. a specific number.

In a second step, the CityObjectGroup objects representing different storeys are grouped themselves. By using the generic aggregation concept of CityObjectGroup, the “storeys group” is associated with the corresponding Building or BuildingPart object. The class attribute of the storeys group shall be assigned the value “building storeys”.

12.6.37. Requirements

Requirement 9	**/req/building/base**
A	If a building only consists of one (homogeneous) part, it shall be represented by the element Building. However, if a building is composed of individual structural segments, it shall be modelled as a Building element having one or more additional BuildingPart elements. Only the geometry and non-spatial prop-erties of the main part of the building should be represented within the aggregating Building element.

Requirement 10	**/req/building/refIntegrity**

A	The boundedBy property (type: BoundarySurfacePropertyType) of the element _AbstractBuilding may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface ele-ment using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and Clo-sureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of _AbstractBuilding.
B	The outerBuildingInstallation property (type: BuildingInstallationPropertyType) of the element _AbstractBuilding may contain a BuildingInstallation element inline or an XLink reference to a remote BuildingInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href at-tribute of the outerBuildingInstallation property may only point to a remote BuildingInstallation ele-ment (where remote BuildingInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
C	The interiorBuildingInstallation property (type: IntBuildingInstallationPropertyType) of the element _AbstractBuilding may contain an IntBuildingInstallation element inline or an XLink reference to a re-mote IntBuildingInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorBuildingInstallation property may only point to a remote IntBuildingInstallation element (where remote IntBuildingInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
D	The interiorRoom property (type: InteriorRoomPropertyType) of the element _AbstractBuilding may contain a Room element inline or an XLink reference to a remote Room element using the XLink con-cept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorRoom property may only point to a remote Room element (where remote Room elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but nei-ther both nor none.

E	The consistsOfBuildingPart property (type: BuildingPartPropertyType) of the element _AbstractBuilding may contain a BuildingPart element inline or an XLink reference to a remote Build-ingPart element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the consistsOfBuildingPart property may only point to a remote BuildingPart element (where remote BuildingPart elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
F	The address property (type: core:AddressPropertyType) of the element _AbstractBuilding may contain an core:Address element inline or an XLink reference to a remote core:Address element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the address property may on-ly point to a remote core:Address element (where remote core:Address elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
G	The opening property (type: OpeningPropertyType) of the element _BoundarySurface may contain an _Opening element inline or an XLink reference to a remote _Opening element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the opening property may only point to a re-mote _Opening element (where remote _Opening elements are located in another document or else-where in the same document). Either the contained element or the reference must be given, but neither both nor none.
H	The address property (type: core:AddressPropertyType) of the element Door may contain an core:Address element inline or an XLink reference to a remote core:Address element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the address property may only point to a remote core:Address element (where remote core:Address elements are located in another docu-ment or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

I	The boundedBy property (type: BoundarySurfacePropertyType) of the element BuildingInstallation may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the bound-edBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface elements are located in another document or elsewhere in the same document). Either the contained el-ement or the reference must be given, but neither both nor none. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and Clo-sureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of Build-ingInstallation.
J	The boundedBy property (type: BoundarySurfacePropertyType) of the element IntBuildingInstallation may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the bound-edBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface elements are located in another document or elsewhere in the same document). Either the contained el-ement or the reference must be given, but neither both nor none. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of IntBuildingInstallation.
K	The boundedBy property (type: BoundarySurfacePropertyType) of the element Room may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of Room.

L	The interiorFurniture property (type: InteriorFurniturePropertyType) of the element Room may con-tain an BuildingFurniture element inline or an XLink reference to a remote BuildingFurniture element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorFurni-ture property may only point to a remote BuildingFurniture element (where remote BuildingFurniture elements are located in another document or elsewhere in the same document). Either the contained el-ement or the reference must be given, but neither both nor none.
M	The roomInstallation property (type: IntBuildingInstallationPropertyType) of the element Room may contain an IntBuildingInstallation element inline or an XLink reference to a remote IntBuildingInstalla-tion element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the roomInstallation property may only point to a remote IntBuildingInstallation element (where remote IntBuildingInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
N	The lodXImplicitRepresentation, X [2..4], property (type: core:ImplicitRepresentationPropertyType) of the element BuildingInstallation may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the lat-ter case, the xlink:href attribute of the lodXImplicitRepresentation, X [2..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
O	The lod4ImplicitRepresentation property (type: core:ImplicitRepresentationPropertyType) of the ele-ment IntBuildingInstallation may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lod4ImplicitRepresentation property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

P	The lodXImplicitRepresentation, X □ [3..4], property (type: core:ImplicitRepresentationPropertyType) of the element _Opening may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lodXImplicitRepresentation, X □ [3..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
Q	The lod4ImplicitRepresentation property (type: core:ImplicitRepresentationPropertyType) of the ele-ment BuildingFurniture may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lod4ImplicitRepresentation property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

Requirement 11	**/req/building/restrictions**
A	The gml:MultiSurface geometries that are associated using the lod0FootPrint and lod0RoofEdge prop-erties must have 3D coordinates. For each surface, the height values of the coordinate tuples belonging to the same surface shall be identical.
B	The lodXSolid and lodXMultiSurface, X □ [1..4], properties (gml:SolidPropertyType resp. gml:MultiSurfacePropertyType) of _AbstractBuilding may be used to geometricllay represent the exte-rior shell of a building (as volume or surface model) within each LOD. For LOD1, either lod1Solid or lod1MultiSurface must be used, but not both. Starting from LOD2, both properties may be modelled in-dividually and complementary.

C	Starting from LOD2, the exterior shell of an _AbstractBuilding may be semantically decomposed into _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of _AbstractBuilding. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloor-Surface and ClosureSurface as subclasses of _BoundarySurface are allowed. The boundedBy property (not to be confused with the gml:boundedBy property) shall not be used if the building is only repre-sented in LOD1. If the exterior shell is represented by _BoundarySurface elements, an additional geometric representa-tion as volume or surface model using the lodXSolid and lodXMultiSurface, X ▯ [2..4], properties shall not explicitly define the geometry, but has to reference the according components of the gml:MultiSurface element of _BoundarySurface within each LOD using the XLink concept of GML 3.1.1.
D	Starting from LOD2, curve parts of the building shell may be represented using the lodXMultiCurve, X ▯ [2..4], property of _AbstractBuilding. This property shall not be used if the building is only repre-sented in LOD1.
E	Starting from LOD2, the outerBuildingInstallation property (type: BuildingInstallationPropertyType) of _AbstractBuilding may be used to model BuildingInstallation elements. BuildingInstallation elements shall only be used to represent outer characteristics of a building which do not have the significance of building parts. The outerBuildingInstallation property shall not be used if the building is only repre-sented in LOD1.
F	Starting from LOD2, the geometry of BuildingInstallation elements may be semantically classified by _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of BuildingInstallation. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, Outer-FloorSurface and ClosureSurface as subclasses of _BoundarySurface are allowed.

G	Starting from LOD3, openings of _BoundarySurface elements may be modelled using the opening property (type: OpeningPropertyType) of _BoundarySurface. This property shall not be used for _BoundarySurface elements only represented in LOD2. Accordingly, the surface geometry representing a _BoundarySurface in LOD2 must be simply connected. The opening property of _BoundarySurface may contain or reference _Opening elements. If the geo-metric location of an _Opening element topologically lies within a surface component of the _BoundarySurface, the opening must also be represented as inner hole of that surface. The embrasure surface of an _Opening element shall belong to the relevant adjacent _BoundarySurface.
G	Starting from LOD4, the interiorRoom property (type: InteriorRoomPropertyType) of _AbstractBuilding may be used to semantically model the free space inside the building by Room ele-ments. This property shall not be used if the building is only represented in LOD 1 – 3. The Room ele-ment may be geometrically represented as a surface or volume model, using its lod4Solid or lod4MultiSurface property (gml:SolidPropertyType resp. gml:MultiSurfacePropertyType). In addition, different parts of the visible surface of a room may be modelled by thematic _BoundarySurface elements. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSur-face as subclasses of _BoundarySurface are allowed. If the visible surface of a room is represented by _BoundarySurface elements, an additional geometric representation as volume or surface model using the lod4Solid and lod4MultiSurface property shall not explicitly define the geometry, but has to refer-ence the according components of the gml:MultiSurface element of _BoundarySurface using the XLink concept of GML 3.1.1.
H	Starting from LOD4, the interiorBuildingInstallation property (type: IntBuildingInstallationProperty-Type) of _AbstractBuilding may be used to represent immovable objects inside the building that are permamently attached to the building structure. The interiorBuildingInstallation property shall not be used if the building is only represented in LOD 1 – 3. Furthermore, the interiorBuildingInstallation property shall only be used if the object cannot be associated with a Room element. In the latter case, the roomInstallation property (type: IntBuildingInstallationPropertyType) of the corresponding Room element shall be used to represent the object.

I	Starting from LOD4, the geometry of IntBuildingInstallation elements may be semantically classified by _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of IntBuildingInstallation. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface as subclasses of _BoundarySurface are allowed.

12.7. Tunnel

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-tunnel	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Tunnel Model is depicted in Tunnel UML Diagram. The Data Dictionary for the Tunnel Package is provided in section Tunnel Model Data Dictionary.

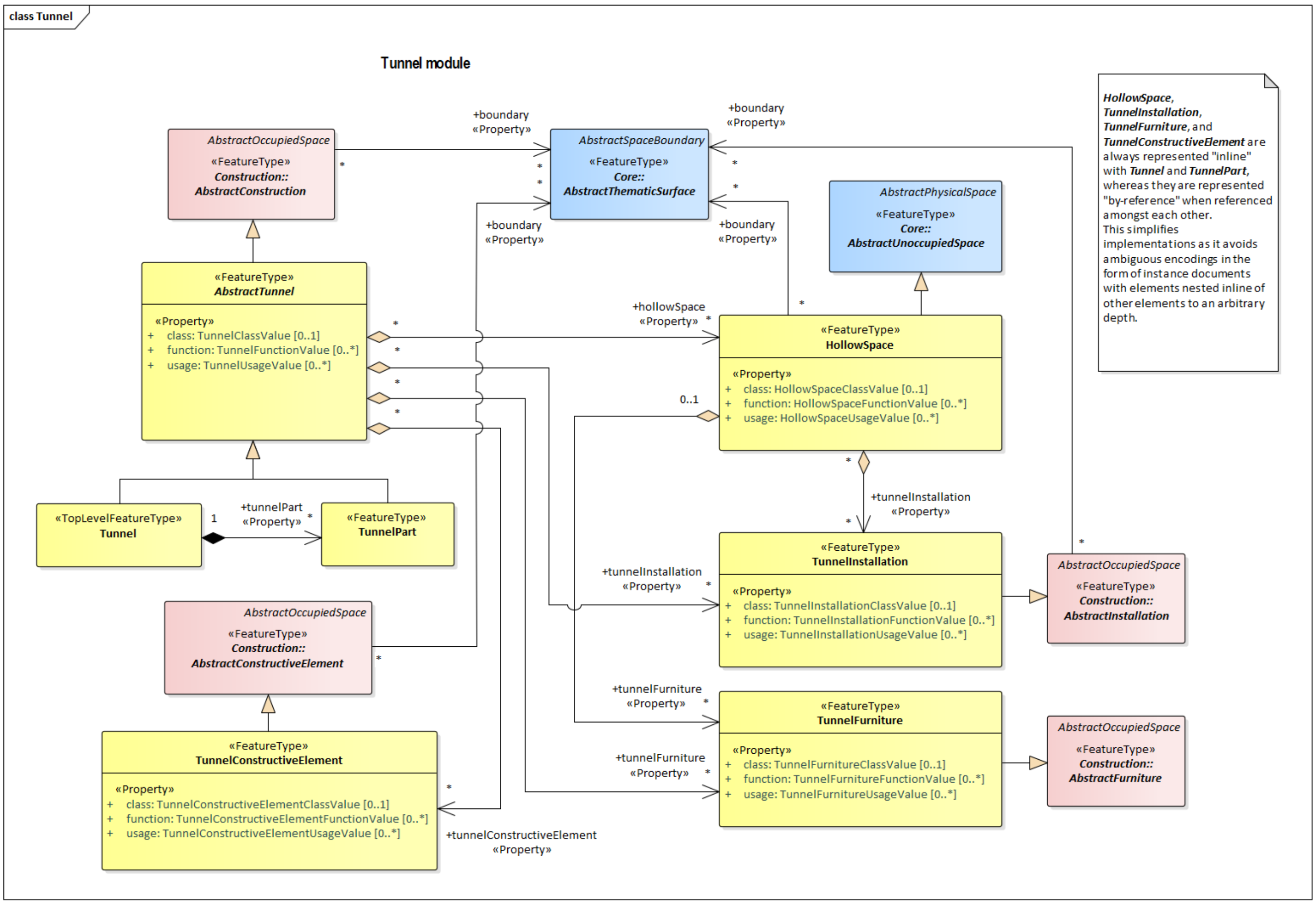


Figure 18. UML diagram of the Tunnel Model.

The UML diagram of the Tunnel Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.8. Tunnel Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.8.1. Class AbstractTunnel

Subclass of AbstractConstruction

Class AbstractTunnel
Definition: Subtype Of: AbstractConstruction StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: tunnelFurniture Target Class: TunnelFurniture
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: Target Class: AbstractConstruction
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: tunnelInstallation Target Class: TunnelInstallation

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTunnel
Target Role:	tunnelConstructiveElement
Target Class:	TunnelConstructiveElement
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTunnel
Target Role:	hollowSpace
Target Class:	HollowSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TunnelPart
Target Role:	
Target Class:	AbstractTunnel
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Tunnel
Target Role:	
Target Class:	AbstractTunnel

Attributes

Attribute Name: class
Value Type: TunnelClassValue
Definition: SIG3D: Classification of the actual usage of Tunnel or TunnelPart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: TunnelFunctionValue
Definition: SIG3D: Specified function of Tunnel or TunnelPart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: usage Value Type: TunnelUsageValue Definition: SIG3D: Actual usage of Tunnel or TunnelPart as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.8.2. Class HollowSpace

Subclass of AbstractUnoccupiedSpace

Class HollowSpace
Definition: Subtype Of: AbstractUnoccupiedSpace StereoType: «FeatureType»

Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: HollowSpace Target Role: boundary Target Class: AbstractThematicSurface
Name: Type: Association Direction: Source → Destination Source Role: Source Class: HollowSpace Target Role: tunnelFurniture Target Class: TunnelFurniture
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: HollowSpace Target Role: Target Class: AbstractUnoccupiedSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: HollowSpace Target Role: tunnelInstallation Target Class: TunnelInstallation

Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: hollowSpace Target Class: HollowSpace
Attributes
Attribute Name: class Value Type: HollowSpaceClassValue Definition: SIG3D: Classification of HollowSpace as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: HollowSpaceFunctionValue Definition: SIG3D: Specified function of HollowSpace as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: HollowSpaceUsageValue Definition: SIG3D: Actual usage of HollowSpace as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.8.3. Class HollowSpaceClassValue

Subclass of <←section,>>

Class HollowSpaceClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.4. Class HollowSpaceFunctionValue

Subclass of <←section,>>

Class HollowSpaceFunctionValue

Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.5. Class HollowSpaceUsageValue

Subclass of <←section,>>

Class HollowSpaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.6. Class Tunnel

Subclass of AbstractTunnel

Class Tunnel
Definition: Subtype Of: AbstractTunnel StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Tunnel Target Role: Target Class: AbstractTunnel
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Tunnel Target Role: tunnelPart Target Class: TunnelPart
Attributes

12.8.7. Class TunnelClassValue

Subclass of <←section,>>

Class TunnelClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.8. Class TunnelConstructiveElement

Subclass of AbstractConstructiveElement

Class TunnelConstructiveElement
Definition: Subtype Of: AbstractConstructiveElement StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelConstructiveElement Target Role: Target Class: AbstractConstructiveElement
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: tunnelConstructiveElement Target Class: TunnelConstructiveElement
Attributes
Attribute Name: class Value Type: TunnelConstructiveElementClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»

Attribute Name: function Value Type: TunnelConstructiveElementFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: TunnelConstructiveElementUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.8.9. Class TunnelConstructiveElementClassValue

Subclass of <←section,>>

Class TunnelConstructiveElementClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.10. Class TunnelConstructiveElementFunctionValue

Subclass of <←section,>>

Class TunnelConstructiveElementFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.11. Class TunnelConstructiveElementUsageValue

Subclass of <←section,>>

Class TunnelConstructiveElementUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»

Associations
Attributes

12.8.12. Class TunnelFunctionValue

Subclass of <←section,>>

Class TunnelFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.13. Class TunnelFurniture

Subclass of AbstractFurniture

Class TunnelFurniture
Definition: Subtype Of: AbstractFurniture StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelFurniture Target Role: Target Class: AbstractFurniture
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: tunnelFurniture Target Class: TunnelFurniture

Name: Type: Association Direction: Source → Destination Source Role: Source Class: HollowSpace Target Role: tunnelFurniture Target Class: TunnelFurniture
Attributes
Attribute Name: class Value Type: TunnelFurnitureClassValue Definition: SIG3D: Classification of TunnelFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: TunnelFurnitureFunctionValue Definition: SIG3D: Specified function of TunnelFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: TunnelFurnitureUsageValue Definition: SIG3D: Actual usage of TunnelFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.8.14. Class TunnelFurnitureClassValue

Subclass of <←section,>>

Class TunnelFurnitureClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.15. Class TunnelFurnitureFunctionValue

Subclass of <←section,>>

Class TunnelFurnitureFunctionValue

Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.16. Class TunnelFurnitureUsageValue

Subclass of <←section,>>

Class TunnelFurnitureUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.17. Class TunnelInstallation

Subclass of AbstractInstallation

Class TunnelInstallation
Definition: Subtype Of: AbstractInstallation StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelInstallation Target Role: Target Class: AbstractInstallation
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTunnel Target Role: tunnelInstallation Target Class: TunnelInstallation

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: HollowSpace
Target Role: tunnelInstallation
Target Class: TunnelInstallation

Attributes

Attribute Name: class
Value Type: TunnelInstallationClassValue
Definition: SIG3D: Classification of TunnelInstallation as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: TunnelInstallationFunctionValue
Definition: SIG3D: Specified function of TunnelInstallation as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: usage
Value Type: TunnelInstallationUsageValue
Definition: SIG3D: Actual usage of TunnelInstallation as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

12.8.18. Class TunnelInstallationClassValue

Subclass of <←section,>>

Class TunnelInstallationClassValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.8.19. Class TunnelInstallationFunctionValue

Subclass of <←section,>>

Class TunnelInstallationFunctionValue

Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.20. Class TunnelInstallationUsageValue

Subclass of <←section,>>

Class TunnelInstallationUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.21. Class TunnelPart

Subclass of AbstractTunnel

Class TunnelPart
Definition: Subtype Of: AbstractTunnel StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelPart Target Role: Target Class: AbstractTunnel
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Tunnel Target Role: tunnelPart Target Class: TunnelPart
Attributes

12.8.22. Class TunnelUsageValue

Subclass of <←section,>>

Class TunnelUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.8.23. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.8.24. Requirements

Requirement 12	/req/tunnel/base
A	If a tunnel only consists of one (homogeneous) part, it shall be represented by the element Tunnel. However, if a tunnel is composed of individual structural segments, it shall be modelled as a Tunnel el-ement having one or more additional TunnelPart elements. Only the geometry and non-spatial proper-ties of the main part of the tunnel should be represented within the aggregating Tunnel element.

Requirement 13	/req/tunnel/refIntegrity

A	The boundedBy property (type: BoundarySurfacePropertyType) of the element _AbstractTunnel may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface ele-ment using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and Clo-sureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of _AbstractTunnel.
B	The outerTunnelInstallation property (type: TunnelInstallationPropertyType) of the element _AbstractTunnel may contain a TunnelInstallation element inline or an XLink reference to a remote TunnelInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href at-tribute of the outerTunnelInstallation property may only point to a remote TunnelInstallation element (where remote TunnelInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
C	The interiorTunnelInstallation property (type: IntTunnelInstallationPropertyType) of the element _AbstractTunnel may contain an IntTunnelInstallation element inline or an XLink reference to a remote IntTunnelInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorTunnelInstallation property may only point to a remote IntTunnelInstallation el-ement (where remote IntTunnelInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
D	The interiorHollowSpace property (type: InteriorHollowSpacePropertyType) of the element _AbstractTunnel may contain a HollowSpace element inline or an XLink reference to a remote Hol-lowSpace element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorHollowSpace property may only point to a remote HollowSpace element (where remote Hol-lowSpace elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

E	The consistsOfTunnelPart property (type: TunnelPartPropertyType) of the element _AbstractTunnel may contain a TunnelPart element inline or an XLink reference to a remote TunnelPart element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the consistsOfTunnelPart property may only point to a remote TunnelPart element (where remote TunnelPart elements are locat-ed in another document or elsewhere in the same document). Either the contained element or the refer-ence must be given, but neither both nor none.
F	The opening property (type: OpeningPropertyType) of the element _BoundarySurface may contain an _Opening element inline or an XLink reference to a remote _Opening element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the opening property may only point to a remote _Opening element (where remote _Opening elements are located in another document or else-where in the same document). Either the contained element or the reference must be given, but neither both nor none.
G	The boundedBy property (type: BoundarySurfacePropertyType) of the element TunnelInstallation may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface ele-ment using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and Clo-sureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of Tun-nelInstallation.

H	The boundedBy property (type: BoundarySurfacePropertyType) of the element IntTunnelInstallation may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface elements are located in another document or elsewhere in the same document). Either the contained el-ement or the reference must be given, but neither both nor none. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of IntTunnelInstallation.
I	The boundedBy property (type: BoundarySurfacePropertyType) of the element HollowSpace may con-tain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of HollowSpace.
J	The interiorFurniture property (type: InteriorFurniturePropertyType) of the element HollowSpace may contain an TunnelFurniture element inline or an XLink reference to a remote TunnelFurniture element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorFurni-ture property may only point to a remote TunnelFurniture element (where remote TunnelFurniture elements are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none.

K	The hollowSpaceInstallation property (type: IntTunnelInstallationPropertyType) of the element Hol-lowSpace may contain an IntTunnelInstallation element inline or an XLink reference to a remote IntTunnelInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the hollowSpaceInstallation property may only point to a remote IntTunnelInstallation ele-ment (where remote IntTunnelInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
L	The lodXImplicitRepresentation, X □ [2..4], property (type: core:ImplicitRepresentationPropertyType) of the element TunnelInstallation may contain a core:ImplicitGeometry element inline or an XLink ref-erence to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lodXImplicitRepresentation, X □ [2..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
M	The lod4ImplicitRepresentation property (type: core:ImplicitRepresentationPropertyType) of the ele-ment IntTunnelInstallation may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lod4ImplicitRepresentation property may only point to a remotecore:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
N	The lodXImplicitRepresentation, X □ [3..4], property (type: core:ImplicitRepresentationPropertyType) of the element _Opening may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lodXImplicitRepresentation, X □ [3..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

O	The lod4ImplicitRepresentation property (type: core:ImplicitRepresentationPropertyType) of the ele-ment TunnelFurniture may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lod4ImplicitRepresentation property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

Requirement 14	**/req/tunnel/restrictions**
A	The lodXSolid and lodXMultiSurface, X ∈ [1..4], properties (gml:SolidPropertyType resp. gml:MultiSurfacePropertyType) of _AbstractTunnel may be used to geometrically represent the exterior shell of a tunnel (as volume or surface model) within each LOD. For LOD1, either lod1Solid or lod1MultiSurface must be used, but not both. Starting from LOD2, both properties may be modelled in-dividually and complementary.
B	Starting from LOD2, the exterior shell of an _AbstractTunnel may be semantically decomposed into _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of _AbstractTunnel. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloor-Surface and ClosureSurface as subclasses of _BoundarySurface are allowed. The boundedBy property (not to be confused with the gml:boundedBy property) shall not be used if the tunnel is only represented in LOD1. If the exterior shell is represented by _BoundarySurface elements, an additional geometric representa-tion as volume or surface model using the lodXSolid and lodXMultiSurface, X ∈ [2..4], properties shall not explicitly define the geometry, but has to reference the according components of the gml:MultiSurface element of _BoundarySurface within each LOD using the XLink concept of GML 3.1.1.
C	Starting from LOD2, curve parts of the tunnel shell may be represented using the lodXMultiCurve, X ∈ [2..4], property of _AbstractTunnel. This property shall not be used if the tunnel is only represented in LOD1.

D	Starting from LOD2, the outerTunnelInstallation property (type: TunnelInstallationPropertyType) of _AbstractTunnel may be used to model TunnelInstallation elements. TunnelInstallation elements shall only be used to represent outer characteristics of a tunnel which do not have the significance of tunnel parts. The outerTunnelInstallation property shall not be used if the tunnel is only represented in LOD1.
E	Starting from LOD2, the geometry of TunnelInstallation elements may be semantically classified by _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of TunnelInstallation. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloor-Surface and ClosureSurface as subclasses of _BoundarySurface are allowed.
F	Starting from LOD3, openings of _BoundarySurface elements may be modelled using the opening property (type: OpeningPropertyType) of _BoundarySurface. This property shall not be used for _BoundarySurface elements only represented in LOD2. Accordingly, the surface geometry representing a _BoundarySurface in LOD2 must be simply connected. The opening property of _BoundarySurface may contain or reference _Opening elements. If the geo-metric location of an _Opening element topologically lies within a surface component of the _BoundarySurface, the opening must also be represented as inner hole of that surface. The embrasure surface of an _Opening element shall belong to the relevant adjacent _BoundarySurface.

G	Starting from LOD4, the interiorHollowSpace property (type: InteriorHollowSpacePropertyType) of _AbstractTunnel may be used to semantically model the free space inside the tunnel by HollowSpace elements. This property shall not be used if the tunnel is only represented in LOD 1 – 3. The Hol-lowSpace element may be geometrically represented as a surface or volume model, using its lod4Solid or lod4MultiSurface property (gml:SolidPropertyType resp. gml:MultiSurfacePropertyType). In addition, different parts of the visible surface of a hollow space may be modelled by thematic _BoundarySurface elements. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSur-face as subclasses of _BoundarySurface are allowed. If the visible surface of a hollow space is repre-sented by _BoundarySurface elements, an additional geometric representation as volume or surface model using the lod4Solid and lod4MultiSurface property shall not explicitly define the geometry, but has to reference the according components of the gml:MultiSurface element of _BoundarySurface using the XLink concept of GML 3.1.1.
H	Starting from LOD4, the interiorTunnelInstallation property (type: IntTunnelInstallationPropertyType) of _AbstractTunnel may be used to represent immovable objects inside the tunnel that are permanently attached to the tunnel structure. The interiorTunnelInstallation property shall not be used if the tunnel is only represented in LOD 1 – 3. Furthermore, the interiorTunnelInstallation property shall only be used if the object cannot be associated with a HollowSpace element. In the latter case, the hollowSpaceInstal-lation property (type: IntTunnelInstallationPropertyType) of the corresponding HollowSpace element shall be used to represent the object.
I	Starting from LOD4, the geometry of IntTunnelInstallation elements may be semantically classified by _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of IntTunnelInstallation. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface as subclasses of _BoundarySurface are allowed.

12.9. Bridge Model

Requirements Class	
http://www.opengis.net/spec/CityGML/3.0/req/req-class-bridge	
Target type	Conceptual Model
Dependency	TBD

Dependency	TBD

TBD

The UML diagram of the Bridge Model is depicted in Bridge UML Diagram. The Data Dictionary for the Bridge Model Package is provided in section Bridge Model Data Dictionary.

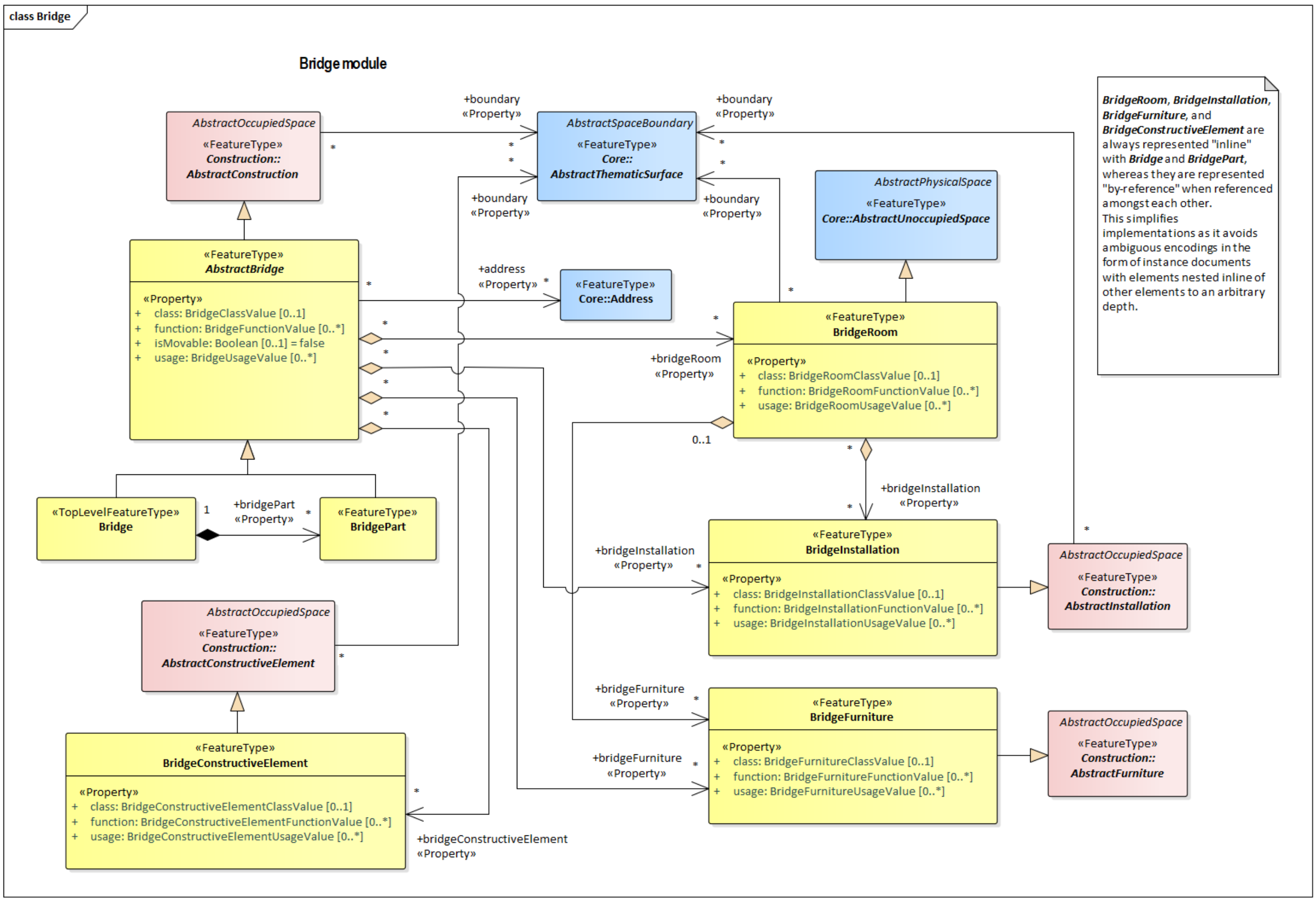


Figure 19. UML diagram of the Bridge Model.

The UML diagram of the Bridge Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.10. Bridge Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.10.1. Class AbstractBridge

Subclass of AbstractConstruction

Class AbstractBridge
Definition: Subtype Of: AbstractConstruction StereoType: «FeatureType»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: Target Class: AbstractConstruction
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: bridgeRoom Target Class: BridgeRoom
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: address Target Class: Address
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: bridgeConstructiveElement Target Class: BridgeConstructiveElement
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: bridgeInstallation Target Class: BridgeInstallation

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: AbstractBridge
Target Role: bridgeFurniture
Target Class: BridgeFurniture

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: Bridge
Target Role:
Target Class: AbstractBridge

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: BridgePart
Target Role:
Target Class: AbstractBridge

Attributes

Attribute Name: class
Value Type: BridgeClassValue
Definition: SIG3D: Classification of the actual usage of Bridge or BridgePart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: BridgeFunctionValue
Definition: SIG3D: Specified function of Bridge or BridgePart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: isMovable
Value Type: Boolean
Definition: SIG3D: Indicates wether a bridge is movable to allow passages for ships (e.g. turnable bridge, liftable bridge)
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: usage
Value Type: BridgeUsageValue
Definition: SIG3D: Actual usage of Bridge or BridgePart as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

12.10.2. Class Bridge

Subclass of AbstractBridge

Class Bridge

Definition:
Subtype Of: AbstractBridge
StereoType: «TopLevelFeatureType»

Associations

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: Bridge
Target Role: bridgePart
Target Class: BridgePart

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: Bridge
Target Role:
Target Class: AbstractBridge

Attributes

12.10.3. Class BridgeClassValue

Subclass of <←section,>>

Class BridgeClassValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.10.4. Class BridgeConstructiveElement

Subclass of AbstractConstructiveElement

Class BridgeConstructiveElement
Definition: Subtype Of: AbstractConstructiveElement StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BridgeConstructiveElement Target Role: Target Class: AbstractConstructiveElement
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: bridgeConstructiveElement Target Class: BridgeConstructiveElement
Attributes
Attribute Name: class Value Type: BridgeConstructiveElementClassValue Definition: SIG3D: Classification of BridgeConstructionElement as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: BridgeConstructiveElementFunctionValue Definition: SIG3D: Specified function of BridgeConstructionElement as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BridgeConstructiveElementUsageValue Definition: SIG3D: Actual usage of BridgeConstructionElement as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.10.5. Class BridgeConstructiveElementClassValue

Subclass of <←section,>>

Class BridgeConstructiveElementClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.6. Class BridgeConstructiveElementFunctionValue

Subclass of <←section,>>

Class BridgeConstructiveElementFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.7. Class BridgeConstructiveElementUsageValue

Subclass of <←section,>>

Class BridgeConstructiveElementUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.8. Class BridgeFunctionValue

Subclass of <←section,>>

Class BridgeFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»

Associations
Attributes

12.10.9. Class BridgeFurniture

Subclass of AbstractFurniture

Class BridgeFurniture
Definition: Subtype Of: AbstractFurniture StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BridgeFurniture Target Role: Target Class: AbstractFurniture
Name: Type: Association Direction: Source → Destination Source Role: Source Class: BridgeRoom Target Role: bridgeFurniture Target Class: BridgeFurniture
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractBridge Target Role: bridgeFurniture Target Class: BridgeFurniture
Attributes
Attribute Name: class Value Type: BridgeFurnitureClassValue Definition: SIG3D: Classification of BridgeFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»

Attribute Name: function Value Type: BridgeFurnitureFunctionValue Definition: SIG3D: Specified function of BridgeFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BridgeFurnitureUsageValue Definition: SIG3D: Actual usage of BridgeFurniture as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.10.10. Class BridgeFurnitureClassValue

Subclass of <←section,>>

Class BridgeFurnitureClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.11. Class BridgeFurnitureFunctionValue

Subclass of <←section,>>

Class BridgeFurnitureFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.12. Class BridgeFurnitureUsageValue

Subclass of <←section,>>

Class BridgeFurnitureUsageValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.10.13. Class BridgeInstallation

Subclass of AbstractInstallation

Class BridgeInstallation

Definition:
Subtype Of: AbstractInstallation
StereoType: «FeatureType»

Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeInstallation
Target Role:	
Target Class:	AbstractInstallation

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	bridgeInstallation
Target Class:	BridgeInstallation

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBridge
Target Role:	bridgeInstallation
Target Class:	BridgeInstallation

Attributes

Attribute Name: class Value Type: BridgeInstallationClassValue Definition: SIG3D: Classification of BridgeInstallation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: BridgeInstallationFunctionValue Definition: SIG3D: Specified function of BridgeInstallation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BridgeInstallationUsageValue Definition: SIG3D: Actual usage of BridgeInstallation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.10.14. Class BridgeInstallationClassValue

Subclass of <←section,>>

Class BridgeInstallationClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.15. Class BridgeInstallationFunctionValue

Subclass of <←section,>>

Class BridgeInstallationFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.16. Class BridgeInstallationUsageValue

Subclass of <←section,>>

Class BridgeInstallationUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.17. Class BridgePart

Subclass of AbstractBridge

Class BridgePart
Definition: Subtype Of: AbstractBridge StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BridgePart Target Role: Target Class: AbstractBridge
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Bridge Target Role: bridgePart Target Class: BridgePart
Attributes

12.10.18. Class BridgeRoom

Subclass of AbstractUnoccupiedSpace

Class BridgeRoom

Definition:	
Subtype Of:	AbstractUnoccupiedSpace
StereoType:	«FeatureType»

Associations

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	bridgeInstallation
Target Class:	BridgeInstallation

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	boundary
Target Class:	AbstractThematicSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	
Target Class:	AbstractUnoccupiedSpace

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeRoom
Target Role:	bridgeFurniture
Target Class:	BridgeFurniture

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBridge
Target Role:	bridgeRoom
Target Class:	BridgeRoom

Attributes

Attribute Name: class Value Type: BridgeRoomClassValue Definition: SIG3D: Classification of BridgeRoom as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: BridgeRoomFunctionValue Definition: SIG3D: Specified function of BridgeRoom as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: BridgeRoomUsageValue Definition: SIG3D: Actual usage of BridgeRoom as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.10.19. Class BridgeRoomClassValue

Subclass of <←section,>>

Class BridgeRoomClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.20. Class BridgeRoomFunctionValue

Subclass of <←section,>>

Class BridgeRoomFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.21. Class BridgeRoomUsageValue

Subclass of <←section,>>

Class BridgeRoomUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.22. Class BridgeUsageValue

Subclass of <←section,>>

Class BridgeUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.10.23. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.10.24. Requirements

Requirement 15	**/req/bridge/base**
A	If a bridge only consists of one (homogeneous) part, it shall be represented by the element Bridge. However, if a bridge is composed of individual structural segments, it shall be modelled as a Bridge el-ement having one or more additional BridgePart elements. Only the geometry and non-spatial proper-ties of the main part of the bridge should be represented within the aggregating Bridge element.

Requirement 16	**/req/bridge/refIntegrity**

A	The boundedBy property (type: BoundarySurfacePropertyType) of the element _AbstractBridge may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface ele-ment using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and Clo-sureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of _AbstractBridge.
B	he outerBridgeConstruction property (type: BridgeConstructionElementPropertyType) of the element _AbstractBridge may contain a BridgeConstructionElement element inline or an XLink reference to a remote BridgeConstructionElement element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the outerBridgeConstruction property may only point to a remote BridgeCon-structionElement element (where remote BridgeConstructionElement elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
C	The outerBridgeInstallation property (type: BridgeInstallationPropertyType) of the element _AbstractBridge may contain a BridgeInstallation element inline or an XLink reference to a remote BridgeInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href at-tribute of the outerBridgeInstallation property may only point to a remote BridgeInstallation element (where remote BridgeInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
D	The interiorBridgeInstallation property (type: IntBridgeInstallationPropertyType) of the element _AbstractBridge may contain an IntBridgeInstallation element inline or an XLink reference to a remote IntBridgeInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorBridgeInstallation property may only point to a remote IntBridgeInstallation el-ement (where remote IntBridgeInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

E	The interiorBridgeRoom property (type: InteriorBridgeRoomPropertyType) of the element _AbstractBridge may contain a BridgeRoom element inline or an XLink reference to a remote BridgeRoom element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorBridgeRoom property may only point to a remote BridgeRoom element (where remote BridgeRoom elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
F	The consistsOfBridgePart property (type: BridgePartPropertyType) of the element _AbstractBridge may contain a BridgePart element inline or an XLink reference to a remote BridgePart element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the consistsOfBridgePart property may only point to a remote BridgePart element (where remote BridgePart elements are located in another document or elsewhere in the same document). Either the contained element or the refer-ence must be given, but neither both nor none.
G	The address property (type: core:AddressPropertyType) of the element _AbstractBridge may contain an core:Address element inline or an XLink reference to a remote core:Address element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the address property may only point to a remote core:Address element (where remote core:Address elements are located in another docu-ment or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
H	The opening property (type: OpeningPropertyType) of the element _BoundarySurface may contain an _Opening element inline or an XLink reference to a remote _Opening element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the opening property may only point to a re-mote _Opening element (where remote _Opening elements are located in another document or else-where in the same document). Either the contained element or the reference must be given, but neither both nor none.

I	The address property (type: core:AddressPropertyType) of the element Door may contain an core:Address element inline or an XLink reference to a remote core:Address element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the address property may only point to a remote core:Address element (where remote core:Address elements are located in another docu-ment or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
J	The boundedBy property (type: BoundarySurfacePropertyType) of the element BridgeConstruc-tionElement may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href at-tribute of the boundedBy property may only point to a remote _BoundarySurface element (where re-mote _BoundarySurface elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
K	The boundedBy property (type: BoundarySurfacePropertyType) of the element BridgeInstallation may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface ele-ment using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloorSurface and Clo-sureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of BridgeInstallation.
L	The boundedBy property (type: BoundarySurfacePropertyType) of the element IntBridgeInstallation may contain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the bound-edBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface elements are located in another document or elsewhere in the same document). Either the contained el-ement or the reference must be given, but neither both nor none. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of IntBridgeInstallation.

M	The boundedBy property (type: BoundarySurfacePropertyType) of the element BridgeRoom may con-tain a _BoundarySurface element inline or an XLink reference to a remote _BoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _BoundarySurface element (where remote _BoundarySurface ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface elements are allowed to be encapsulated or referenced by the boundedBy property of BridgeRoom.
N	The interiorFurniture property (type: InteriorFurniturePropertyType) of the element BridgeRoom may contain an BridgeFurniture element inline or an XLink reference to a remote BridgeFurniture element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the interiorFurni-ture property may only point to a remote BridgeFurniture element (where remote BridgeFurniture ele-ments are located in another document or elsewhere in the same document). Either the contained ele-ment or the reference must be given, but neither both nor none.
O	The bridgeRoomInstallation property (type: IntBridgeInstallationPropertyType) of the element BridgeRoom may contain an IntBridgeInstallation element inline or an XLink reference to a remote IntBridgeInstallation element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the bridgeRoomInstallation property may only point to a remote IntBridgeInstallation ele-ment (where remote IntBridgeInstallation elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
P	The lodXImplicitRepresentation, X ∈ [1..4], property (type: core:ImplicitRepresentationPropertyType) of the element BridgeConstructionElement may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lodXImplicitRepresentation, X ∈ [1..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

Q	The lodXImplicitRepresentation, X ∈ [2..4], property (type: core:ImplicitRepresentationPropertyType) of the element BridgeInstallation may contain a core:ImplicitGeometry element inline or an XLink ref-erence to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lodXImplicitRepresentation, X ∈ [2..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
R	The lodXImplicitRepresentation, X ∈ [3..4], property (type: core:ImplicitRepresentationPropertyType) of the element _Opening may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lodXImplicitRepresentation, X ∈ [3..4], property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
S	The lod4ImplicitRepresentation property (type: core:ImplicitRepresentationPropertyType) of the ele-ment IntBridgeInstallation may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lod4ImplicitRepresentation property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
T	The lod4ImplicitRepresentation property (type: core:ImplicitRepresentationPropertyType) of the ele-ment BridgeFurniture may contain a core:ImplicitGeometry element inline or an XLink reference to a remote core:ImplicitGeometry element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the lod4ImplicitRepresentation property may only point to a remote core:ImplicitGeometry element (where remote core:ImplicitGeometry elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

Requirement 17	/req/bridge/restrictions
A	The lodXSolid and lodXMultiSurface, X X E [1..4], properties (gml:SolidPropertyType resp. gml:MultiSurfacePropertyType) of _AbstractBridge may be used to geometrically represent the exterior shell of a bridge (as volume or surface model) within each LOD. For LOD1, either lod1Solid or lod1MultiSurface must be used, but not both. Starting from LOD2, both properties may be modelled in-dividually and complementary.
B	Starting from LOD2, the exterior shell of an _AbstractBridge may be semantically decomposed into _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of _AbstractBridge. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloor-Surface and ClosureSurface as subclasses of _BoundarySurface are allowed. The boundedBy property (not to be confused with the gml:boundedBy property) shall not be used if the bridge is only represented in LOD1. If the exterior shell is represented by _BoundarySurface elements, an additional geometric representa-tion as volume or surface model using the lodXSolid and lodXMultiSurface, X □ [2..4], properties shall not explicitly define the geometry, but has to reference the according components of the gml:MultiSurface element of _BoundarySurface within each LOD using the XLink concept of GML 3.1.1.
C	Starting from LOD2, curve parts of the bridge shell may be represented using the lodXMultiCurve, X □ [2..4], property of _AbstractBridge. This property shall not be used if the bridge is only represented in LOD1.
D	Starting from LOD1, the outerBridgeConstruction property (type: BridgeConstructionElementProper-tyType) of _AbstractBridge may be used to model BridgeConstructionElement elements. BridgeCon-structionElement elements shall only be used to represent outer characteristics of a bridge which do not have the significance of bridge parts and are essential from a structural point of view.
E	Starting from LOD2, the geometry of BridgeConstructionElement elements may be semantically classi-fied by _BoundarySurface elements using the boundedBy property (type: BoundarySurfaceProperty-Type) of BridgeConstructionElement. The boundedBy property (not to be confused with the gml:boundedBy property) shall not be used if the bridge construction element is only represented in LOD1.

F	Starting from LOD2, the outerBridgeInstallation property (type: BridgeInstallationPropertyType) of _AbstractBridge may be used to model BridgeInstallation elements. BridgeInstallation elements shall only be used to represent outer characteristics of a bridge which do not have the significance of bridge parts and are not essential from a structural point of view.
G	Starting from LOD2, the geometry of BridgeInstallation elements may be semantically classified by _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of BridgeInstallation. Only RoofSurface, WallSurface, GroundSurface, OuterCeilingSurface, OuterFloor-Surface and ClosureSurface as subclasses of _BoundarySurface are allowed.
H	Starting from LOD3, openings of _BoundarySurface elements may be modelled using the opening property (type: OpeningPropertyType) of _BoundarySurface. This property shall not be used for _BoundarySurface elements only represented in LOD2. Accordingly, the surface geometry representing a _BoundarySurface in LOD2 must be simply connected. The opening property of _BoundarySurface may contain or reference _Opening elements. If the geo-metric location of an _Opening element topologically lies within a surface component of the _BoundarySurface, the opening must also be represented as inner hole of that surface. The embrasure surface of an _Opening element shall belong to the relevant adjacent _BoundarySurface.
I	Starting from LOD4, the interiorBridgeRoom property (type: InteriorBridgeRoomPropertyType) of _AbstractBridge may be used to semantically model the free space inside the bridge by BridgeRoom el-ements. This property shall not be used if the bridge is only represented in LOD 1 – 3. The BridgeRoom element may be geometrically represented as a surface or volume model, using its lod4Solid or lod4MultiSurface property (gml:SolidPropertyType resp. gml:MultiSurfacePropertyType). In addition, different parts of the visible surface of a bridge room may be modelled by thematic _BoundarySurface elements. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSur-face as subclasses of _BoundarySurface are allowed. If the visible surface of a room is represented by _BoundarySurface elements, an additional geometric representation as volume or surface model using the lod4Solid and lod4MultiSurface property shall not explicitly define the geometry, but has to refer-ence the according components of the gml:MultiSurface element of _BoundarySurface using the XLink concept of GML 3.1.1.

J	Starting from LOD4, the interiorBridgeInstallation property (type: IntBridgeInstallationPropertyType) of _AbstractBridge may be used to represent immovable objects inside the bridge that are permamently attached to the bridge structure. The interiorBridgeInstallation property shall not be used if the bridge is only represented in LOD 1 – 3. Furthermore, the interiorBridgeInstallation property shall only be used if the object cannot be associated with a BridgeRoom element. In the latter case, the bridgeRoomInstal-lation property (type: IntBridgeInstallationPropertyType) of the corresponding BridgeRoom element shall be used to represent the object.
K	Starting from LOD4, the geometry of IntBridgeInstallation elements may be semantically classified by _BoundarySurface elements using the boundedBy property (type: BoundarySurfacePropertyType) of IntBridgeInstallation. Only FloorSurface, CeilingSurface, InteriorWallSurface, and ClosureSurface as subclasses of _BoundarySurface are allowed.

12.11. Water Body

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-waterbody	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Water Body Model is depicted in Water Body UML Diagram. The Data Dictionary for the Water Body Package is provided in section Water Body Data Dictionary.

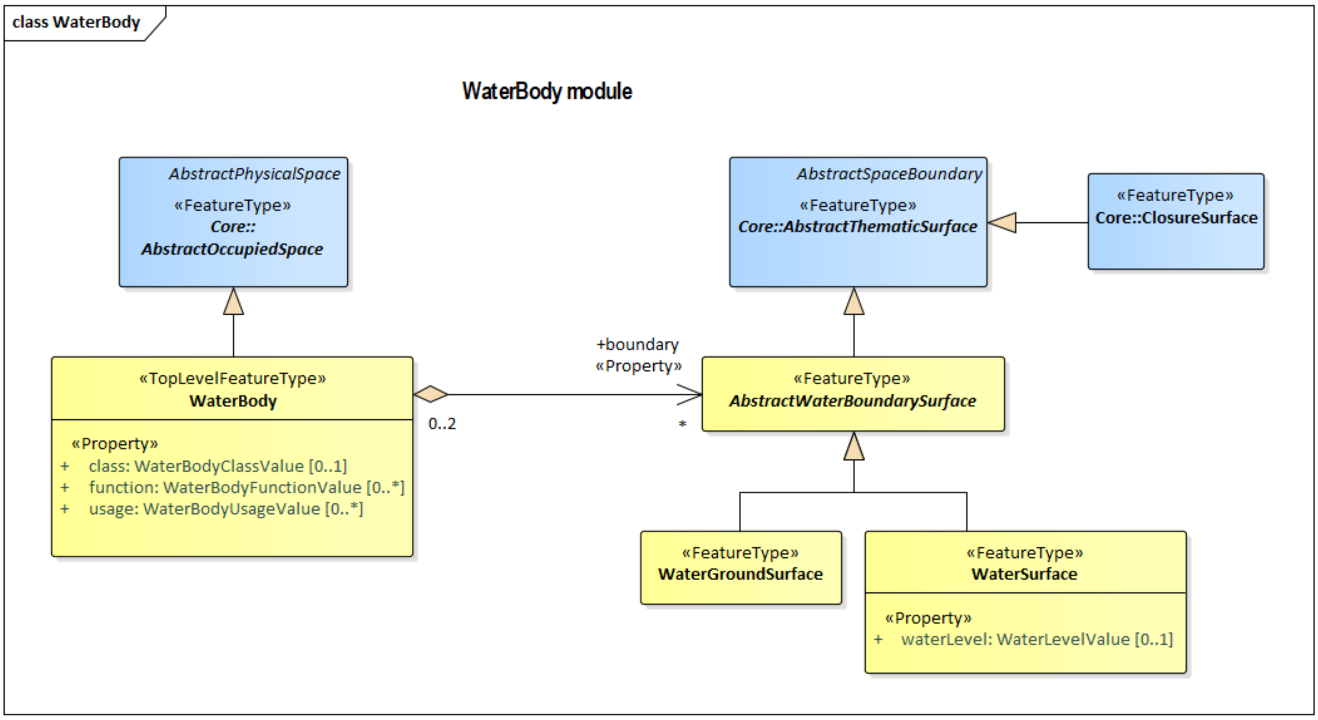


Figure 20. UML diagram of the Water Body Model.

The UML diagram of the Water Body Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.12. WaterBody Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.12.1. Class AbstractWaterBoundarySurface

Subclass of AbstractThematicSurface

Class AbstractWaterBoundarySurface
Definition: Subtype Of: AbstractThematicSurface StereoType: «FeatureType»
Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractWaterBoundarySurface Target Role: Target Class: AbstractThematicSurface
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WaterSurface Target Role: Target Class: AbstractWaterBoundarySurface
Name: Type: Association Direction: Source → Destination Source Role: Source Class: WaterBody Target Role: boundary Target Class: AbstractWaterBoundarySurface
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WaterGroundSurface Target Role: Target Class: AbstractWaterBoundarySurface
Attributes

12.12.2. Class WaterBody

Subclass of AbstractOccupiedSpace

Class WaterBody
Definition: Subtype Of: AbstractOccupiedSpace StereoType: «TopLevelFeatureType»
Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WaterBody Target Role: Target Class: AbstractOccupiedSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: WaterBody Target Role: boundary Target Class: AbstractWaterBoundarySurface
Attributes
Attribute Name: class Value Type: WaterBodyClassValue Definition: SIG3D: Classification of WaterBody as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: WaterBodyFunctionValue Definition: SIG3D: Specified function of WaterBody as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: WaterBodyUsageValue Definition: SIG3D: Actual usage of WaterBody as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.12.3. Class WaterBodyClassValue

Subclass of <←section,>>

Class WaterBodyClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations

Attributes

12.12.4. Class WaterBodyFunctionValue

Subclass of <←section,>>

Class WaterBodyFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.12.5. Class WaterBodyUsageValue

Subclass of <←section,>>

Class WaterBodyUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.12.6. Class WaterGroundSurface

Subclass of AbstractWaterBoundarySurface

Class WaterGroundSurface
Definition: Subtype Of: AbstractWaterBoundarySurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WaterGroundSurface Target Role: Target Class: AbstractWaterBoundarySurface

Attributes

12.12.7. Class WaterLevelValue

Subclass of <←section,>>

Class WaterLevelValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.12.8. Class WaterSurface

Subclass of AbstractWaterBoundarySurface

Class WaterSurface
Definition: Subtype Of: AbstractWaterBoundarySurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WaterSurface Target Role: Target Class: AbstractWaterBoundarySurface
Attributes
Attribute Name: waterLevel Value Type: WaterLevelValue Definition: SIG3D: Codelist of the WaterSurface property waterLevel. Multiplicity: [0..1] Stereotype: «Property»

12.12.9. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.12.10. Requirements

Requirement 18	**/req/waterbody/base**
A	For LOD0 and LOD1, the geometry of a WaterBody may be modelled as a linear network using gml:MultiCurve geometry elements. In that case, each gml:MultiCurve shall be composed of straight line segements, where the line orientation denotes the flow direction. The flow direction is from the first point of a line segment to its last point.
B	Starting from LOD2, the exterior shell of a WaterBody may be semantically decomposed into _WaterBoundarySurface elements using the boundedBy property (type: BoundedByWaterSurfaceProp-ertyType) of WaterBody. The boundedBy property shall not be used if the water body is only represented in lower LODs. If the exterior shell is represented by _WaterBoundarySurface elements, an additional geometric repre-sentation as volume model using the lodXSolid, X [2..4], property of WaterBody shall not explicitly define the geometry, but has to reference the according gml:_Surface elements of the _WaterBoundarySurface objects within each LOD using the XLink concept of GML 3.1.1.
C	Each _WaterBoundarySurface element must have at least one associated surface geometry given by the lodXSurface, X [2..4], properties of _WaterBoundarySurface.
D	_WaterBoundarySurface elements shall only be included as parts of corresponding WaterBody ele-ments. They may not be given as stand-alone city objects within a CityGML model.

Requirement 19	**/req/waterbody/refintegrity**
A	The boundedBy property (type: BoundedByWaterSurfacePropertyType) of the element WaterBody may contain a _WaterBoundarySurface element inline or an XLink reference to a remote _WaterBoundarySurface element using the XLink concept of GML 3.1.1. In the latter case, the xlink:href attribute of the boundedBy property may only point to a remote _WaterBoundarySurface el-ement (where remote _WaterBoundarySurface elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.

12.13. Transportation

Requirements Class	
http://www.opengis.net/spec/CityGML/3.0/req/req-class-transportation	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Transportation Model is depicted in Transportation UML Diagram. The Data Dictionary for the Transportation Package is provided in section Transportation Model Data Dictionary.

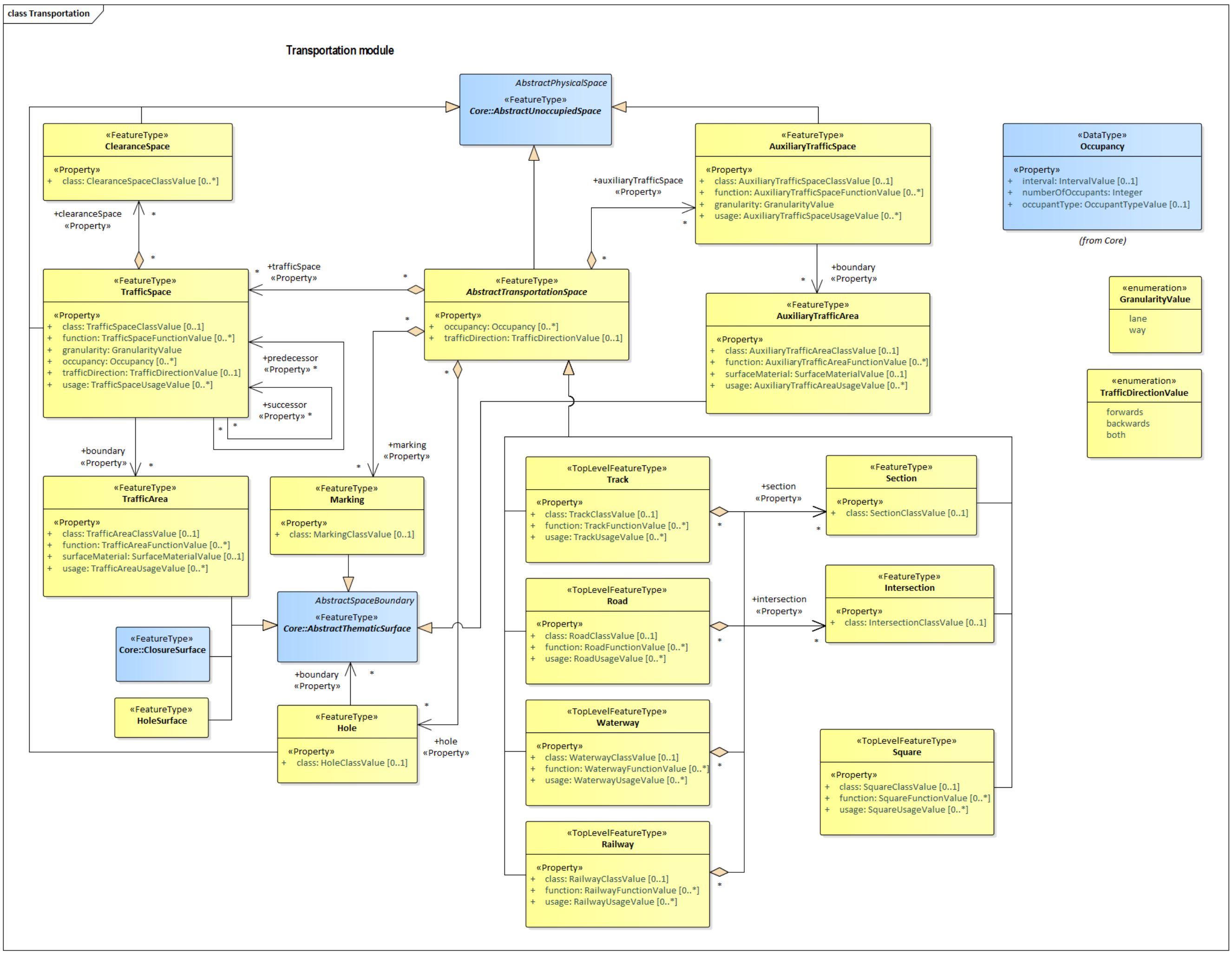


Figure 21. UML diagram of the Transportation Model.

The UML diagram of the Transportation Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.14. Transportation Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.14.1. Class AbstractTransportationSpace

Subclass of AbstractUnoccupiedSpace

Class AbstractTransportationSpace
Definition: Subtype Of: AbstractUnoccupiedSpace StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTransportationSpace Target Role: trafficSpace Target Class: TrafficSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTransportationSpace Target Role: hole Target Class: Hole
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractTransportationSpace Target Role: auxiliaryTrafficSpace Target Class: AuxiliaryTrafficSpace
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractTransportationSpace Target Role: Target Class: AbstractUnoccupiedSpace

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTransportationSpace
Target Role:	marking
Target Class:	Marking

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Intersection
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Waterway
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Track
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Section
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Road
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Railway
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Square
Target Role:	
Target Class:	AbstractTransportationSpace

Attributes

Attribute Name:	occupancy
Value Type:	Occupancy
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

Attribute Name:	trafficDirection
Value Type:	TrafficDirectionValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

12.14.2. Class AuxiliaryTrafficArea

Subclass of AbstractThematicSurface

Class AuxiliaryTrafficArea

Definition:	
Subtype Of:	AbstractThematicSurface
StereoType:	«FeatureType»

Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AuxiliaryTrafficArea
Target Role:	
Target Class:	AbstractThematicSurface

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AuxiliaryTrafficSpace
Target Role:	boundary
Target Class:	AuxiliaryTrafficArea

Attributes

Attribute Name:	class
Value Type:	AuxiliaryTrafficAreaClassValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	function
Value Type:	AuxiliaryTrafficAreaFunctionValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

Attribute Name:	surfaceMaterial
Value Type:	SurfaceMaterialValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	usage
Value Type:	AuxiliaryTrafficAreaUsageValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

12.14.3. Class AuxiliaryTrafficAreaClassValue

Subclass of <←section,>>

Class AuxiliaryTrafficAreaClassValue

Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»

Associations

Attributes

12.14.4. Class AuxiliaryTrafficAreaFunctionValue

Subclass of <←section,>>

Class AuxiliaryTrafficAreaFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.5. Class AuxiliaryTrafficAreaUsageValue

Subclass of <←section,>>

Class AuxiliaryTrafficAreaUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.6. Class AuxiliaryTrafficSpace

Subclass of AbstractUnoccupiedSpace

Class AuxiliaryTrafficSpace
Definition: Subtype Of: AbstractUnoccupiedSpace StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AuxiliaryTrafficSpace Target Role: Target Class: AbstractUnoccupiedSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AuxiliaryTrafficSpace Target Role: boundary Target Class: AuxiliaryTrafficArea

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: AbstractTransportationSpace
Target Role: auxiliaryTrafficSpace
Target Class: AuxiliaryTrafficSpace

Attributes

Attribute Name: class
Value Type: AuxiliaryTrafficSpaceClassValue
Definition: SIG3D: Classification of AuxiliaryTrafficSpace as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: AuxiliaryTrafficSpaceFunctionValue
Definition: SIG3D: Specified function of AuxiliaryTrafficSpace given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: granularity
Value Type: GranularityValue
Definition:
Multiplicity:
Stereotype: «Property»

Attribute Name: usage
Value Type: AuxiliaryTrafficSpaceUsageValue
Definition: SIG3D: Actual usage of AuxiliaryTrafficSpace as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

12.14.7. Class AuxiliaryTrafficSpaceClassValue

Subclass of <←section,>>

Class AuxiliaryTrafficSpaceClassValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.14.8. Class AuxiliaryTrafficSpaceFunctionValue

Subclass of <←section,>>

Class AuxiliaryTrafficSpaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.9. Class AuxiliaryTrafficSpaceUsageValue

Subclass of <←section,>>

Class AuxiliaryTrafficSpaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.10. Class ClearanceSpace

Subclass of AbstractUnoccupiedSpace

Class ClearanceSpace
Definition: Subtype Of: AbstractUnoccupiedSpace StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: ClearanceSpace Target Role: Target Class: AbstractUnoccupiedSpace

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: TrafficSpace
Target Role: clearanceSpace
Target Class: ClearanceSpace

Attributes

Attribute Name: class
Value Type: ClearanceSpaceClassValue
Definition:
Multiplicity: [0..*]
Stereotype: «Property»

12.14.11. Class ClearanceSpaceClassValue

Subclass of <←section,>>

Class ClearanceSpaceClassValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.14.12. Class Hole

Subclass of AbstractUnoccupiedSpace

Class Hole

Definition:
Subtype Of: AbstractUnoccupiedSpace
StereoType: «FeatureType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: Hole
Target Role:
Target Class: AbstractUnoccupiedSpace

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: Hole
Target Role: boundary
Target Class: AbstractThematicSurface

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: AbstractTransportationSpace
Target Role: hole
Target Class: Hole

Attributes

Attribute Name: class
Value Type: HoleClassValue
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

12.14.13. Class HoleClassValue

Subclass of <←section,>>

Class HoleClassValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Name:
Type: NoteLink
Direction: Source → Destination
Source Role:
Source Class: Note
Target Role:
Target Class: HoleClassValue

Attributes

12.14.14. Class HoleSurface

Subclass of AbstractThematicSurface

Class HoleSurface
Definition: Subtype Of: AbstractThematicSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: HoleSurface Target Role: Target Class: AbstractThematicSurface
Attributes

12.14.15. Class Intersection

Subclass of AbstractTransportationSpace

Class Intersection
Definition: Subtype Of: AbstractTransportationSpace StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Intersection Target Role: Target Class: AbstractTransportationSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Waterway Target Role: intersection Target Class: Intersection

Name: Type: Association Direction: Source → Destination Source Role: Source Class: Road Target Role: intersection Target Class: Intersection
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Railway Target Role: intersection Target Class: Intersection
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Track Target Role: intersection Target Class: Intersection
Attributes
Attribute Name: class Value Type: IntersectionClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»

12.14.16. Class IntersectionClassValue

Subclass of <←section,>>

Class IntersectionClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.17. Class Marking

Subclass of AbstractThematicSurface

Class Marking	
Definition:	
Subtype Of:	AbstractThematicSurface
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Marking
Target Role:	
Target Class:	AbstractThematicSurface
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTransportationSpace
Target Role:	marking
Target Class:	Marking

Attributes	
Attribute Name:	class
Value Type:	MarkingClassValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

12.14.18. Class MarkingClassValue

Subclass of <←section,>>

Class MarkingClassValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»

Associations

Attributes

12.14.19. Class Railway

Subclass of AbstractTransportationSpace

Class Railway

Definition: Subtype Of: AbstractTransportationSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Railway Target Role: section Target Class: Section
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Railway Target Role: intersection Target Class: Intersection
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Railway Target Role: Target Class: AbstractTransportationSpace
Attributes
Attribute Name: class Value Type: RailwayClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: RailwayFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: RailwayUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.14.20. Class RailwayClassValue

Subclass of <←section,>>

Class RailwayClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.21. Class RailwayFunctionValue

Subclass of <←section,>>

Class RailwayFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.22. Class RailwayUsageValue

Subclass of <←section,>>

Class RailwayUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.23. Class Road

Subclass of AbstractTransportationSpace

Class Road
Definition: Subtype Of: AbstractTransportationSpace StereoType: «TopLevelFeatureType»

Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Road Target Role: intersection Target Class: Intersection
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Road Target Role: section Target Class: Section
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Road Target Role: Target Class: AbstractTransportationSpace
Attributes
Attribute Name: class Value Type: RoadClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: RoadFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: RoadUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.14.24. Class RoadClassValue

Subclass of <←section,>>

Class RoadClassValue

Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.25. Class RoadFunctionValue

Subclass of <←section,>>

Class RoadFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.26. Class RoadUsageValue

Subclass of <←section,>>

Class RoadUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.27. Class Section

Subclass of AbstractTransportationSpace

Class Section
Definition: Subtype Of: AbstractTransportationSpace StereoType: «FeatureType»
Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	Section
Target Role:	
Target Class:	AbstractTransportationSpace

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Waterway
Target Role:	section
Target Class:	Section

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Railway
Target Role:	section
Target Class:	Section

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Road
Target Role:	section
Target Class:	Section

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Track
Target Role:	section
Target Class:	Section

Attributes

Attribute Name:	class
Value Type:	SectionClassValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

12.14.28. Class SectionClassValue

Subclass of <←section,>>

Class SectionClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.29. Class Square

Subclass of AbstractTransportationSpace

Class Square
Definition: Subtype Of: AbstractTransportationSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Square Target Role: Target Class: AbstractTransportationSpace
Attributes
Attribute Name: class Value Type: SquareClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: SquareFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»

Attribute Name:	usage
Value Type:	SquareUsageValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

12.14.30. Class SquareClassValue

Subclass of <←section,>>

Class SquareClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.31. Class SquareFunctionValue

Subclass of <←section,>>

Class SquareFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.32. Class SquareUsageValue

Subclass of <←section,>>

Class SquareUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.33. Class SurfaceMaterialValue

Subclass of <←section,>>

Class SurfaceMaterialValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.34. Class Track

Subclass of AbstractTransportationSpace

Class Track
Definition: Subtype Of: AbstractTransportationSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Track Target Role: Target Class: AbstractTransportationSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Track Target Role: section Target Class: Section
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Track Target Role: intersection Target Class: Intersection
Attributes

Attribute Name: class Value Type: TrackClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: TrackFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: TrackUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.14.35. Class TrackClassValue

Subclass of <←section,>>

Class TrackClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.36. Class TrackFunctionValue

Subclass of <←section,>>

Class TrackFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.37. Class TrackUsageValue

Subclass of <←section,>>

Class TrackUsageValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.14.38. Class TrafficArea

Subclass of AbstractThematicSurface

Class TrafficArea

Definition:
Subtype Of: AbstractThematicSurface
StereoType: «FeatureType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: TrafficArea
Target Role:
Target Class: AbstractThematicSurface

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: TrafficSpace
Target Role: boundary
Target Class: TrafficArea

Attributes

Attribute Name: class
Value Type: TrafficAreaClassValue
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: TrafficAreaFunctionValue
Definition:
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: surfaceMaterial Value Type: SurfaceMaterialValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: usage Value Type: TrafficAreaUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.14.39. Class TrafficAreaClassValue

Subclass of <←section,>>

Class TrafficAreaClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.40. Class TrafficAreaFunctionValue

Subclass of <←section,>>

Class TrafficAreaFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.41. Class TrafficAreaUsageValue

Subclass of <←section,>>

Class TrafficAreaUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»

Associations
Attributes

12.14.42. Class TrafficSpace

Subclass of AbstractUnoccupiedSpace

Class TrafficSpace	
Definition:	
Subtype Of:	AbstractUnoccupiedSpace
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	TrafficSpace
Target Role:	clearanceSpace
Target Class:	ClearanceSpace
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	TrafficSpace
Target Role:	successor
Target Class:	TrafficSpace
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	TrafficSpace
Target Role:	boundary
Target Class:	TrafficArea
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	TrafficSpace
Target Role:	predecessor
Target Class:	TrafficSpace

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: TrafficSpace
Target Role:
Target Class: AbstractUnoccupiedSpace

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: AbstractTransportationSpace
Target Role: trafficSpace
Target Class: TrafficSpace

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: TrafficSpace
Target Role: successor
Target Class: TrafficSpace

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: TrafficSpace
Target Role: predecessor
Target Class: TrafficSpace

Attributes

Attribute Name: class
Value Type: TrafficSpaceClassValue
Definition: SIG3D: Classification of TrafficSpace as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: TrafficSpaceFunctionValue
Definition: SIG3D: Specified function of TrafficSpace given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: granularity Value Type: GranularityValue Definition: Multiplicity: Stereotype: «Property»
Attribute Name: occupancy Value Type: Occupancy Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: trafficDirection Value Type: TrafficDirectionValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: usage Value Type: TrafficSpaceUsageValue Definition: SIG3D: Actual usage of TrafficSpace as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.14.43. Class TrafficSpaceClassValue

Subclass of <←section,>>

Class TrafficSpaceClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.44. Class TrafficSpaceFunctionValue

Subclass of <←section,>>

Class TrafficSpaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations

Attributes

12.14.45. Class TrafficSpaceUsageValue

Subclass of <←section,>>

Class TrafficSpaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.46. Class TransportationSpaceClassValue

Subclass of <←section,>>

Class TransportationSpaceClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.47. Class TransportationSpaceFunctionValue

Subclass of <←section,>>

Class TransportationSpaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.48. Class TransportationSpaceUsageValue

Subclass of <←section,>>

Class TransportationSpaceUsageValue

Definition:
Subtype Of: <←section,>>
StereoType: «CodeList»

Associations

Attributes

12.14.49. Class Waterway

Subclass of AbstractTransportationSpace

Class Waterway

Definition:
Subtype Of: AbstractTransportationSpace
StereoType: «TopLevelFeatureType»

Associations

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: Waterway
Target Role: section
Target Class: Section

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: Waterway
Target Role: intersection
Target Class: Intersection

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: Waterway
Target Role:
Target Class: AbstractTransportationSpace

Attributes

Attribute Name: class
Value Type: WaterwayClassValue
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function Value Type: WaterwayFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: WaterwayUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.14.50. Class WaterwayClassValue

Subclass of <←section,>>

Class WaterwayClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.51. Class WaterwayFunctionValue

Subclass of <←section,>>

Class WaterwayFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.14.52. Class WaterwayUsageValue

Subclass of <←section,>>

Class WaterwayUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»

Associations
Attributes

12.14.53. Class GranularityValue

Subclass of <←section,>>

Class GranularityValue
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes
Attribute Name: lane Value Type: Definition: Multiplicity: Stereotype:
Attribute Name: way Value Type: Definition: Multiplicity: Stereotype:

12.14.54. Class TrafficDirectionValue

Subclass of <←section,>>

Class TrafficDirectionValue
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes
Attribute Name: forwards Value Type: Definition: Multiplicity: Stereotype:

Attribute Name:	backwards
Value Type:	
Definition:	
Multiplicity:	
Stereotype:	

Attribute Name:	both
Value Type:	
Definition:	
Multiplicity:	
Stereotype:	

12.14.55. Additional Information

The following sections provide additional information which may not be readily available through the UML Model

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.14.56. Requirements

Requirement 20	**/req/transportation/base**
A	For LOD0, the geometry of a TransportationComplex shall be modelled using GML line objects repre-senting the centerline of the transportation complex. The line objects shall establish a linear network. Thus, the lod0Network property (type: gml:GeometricComplexPropertyType) of the element Transpor-tationComplex may only contain or reference an appropriate curve geometry element.
B	Starting from LOD2, the trafficArea property (type: TrafficAreaPropertyType) as well as the auxilia-ryTrafficArea property (type: AuxiliaryTrafficAreaPropertyType) of the element TransportationCom-plex may be used. These properties shall not be used if the transportation complex is only represented in lower LODs.

Requirement 21	**/req/transportation/refInteg**

A	The trafficArea property (type: TrafficAreaPropertyType) of the element TransportationComplex may contain a TrafficArea element inline or an XLink reference to a remote TrafficArea element using the XLink concept of GML 3.1.1. In the latter case, the `xlink:href` attribute of the trafficArea property may only point to a remote TrafficArea element (where remote TrafficArea elements are located in another document or elsewhere in the same document). Either the contained element or the reference must be given, but neither both nor none.
B	The auxiliaryTrafficArea property (type: TrafficAreaPropertyType) of the element TransportationCom-plex may contain an AuxiliaryTrafficArea element inline or an XLink reference to a remote Auxilia-ryTrafficArea element using the XLink concept of GML 3.1.1. In the latter case, the `xlink:href` attribute of the auxiliaryTrafficArea property may only point to a remote AuxiliaryTrafficArea element (where remote AuxiliaryTrafficArea elements are located in another document or elsewhere in the same docu-ment). Either the contained element or the reference must be given, but neither both nor none.

12.15. Vegetation

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-vegetation	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Vegetation Model is depicted in Vegetation UML Diagram. The Data Dictionary for the Vegetation Package is provided in section Vegetation Model Data Dictionary.

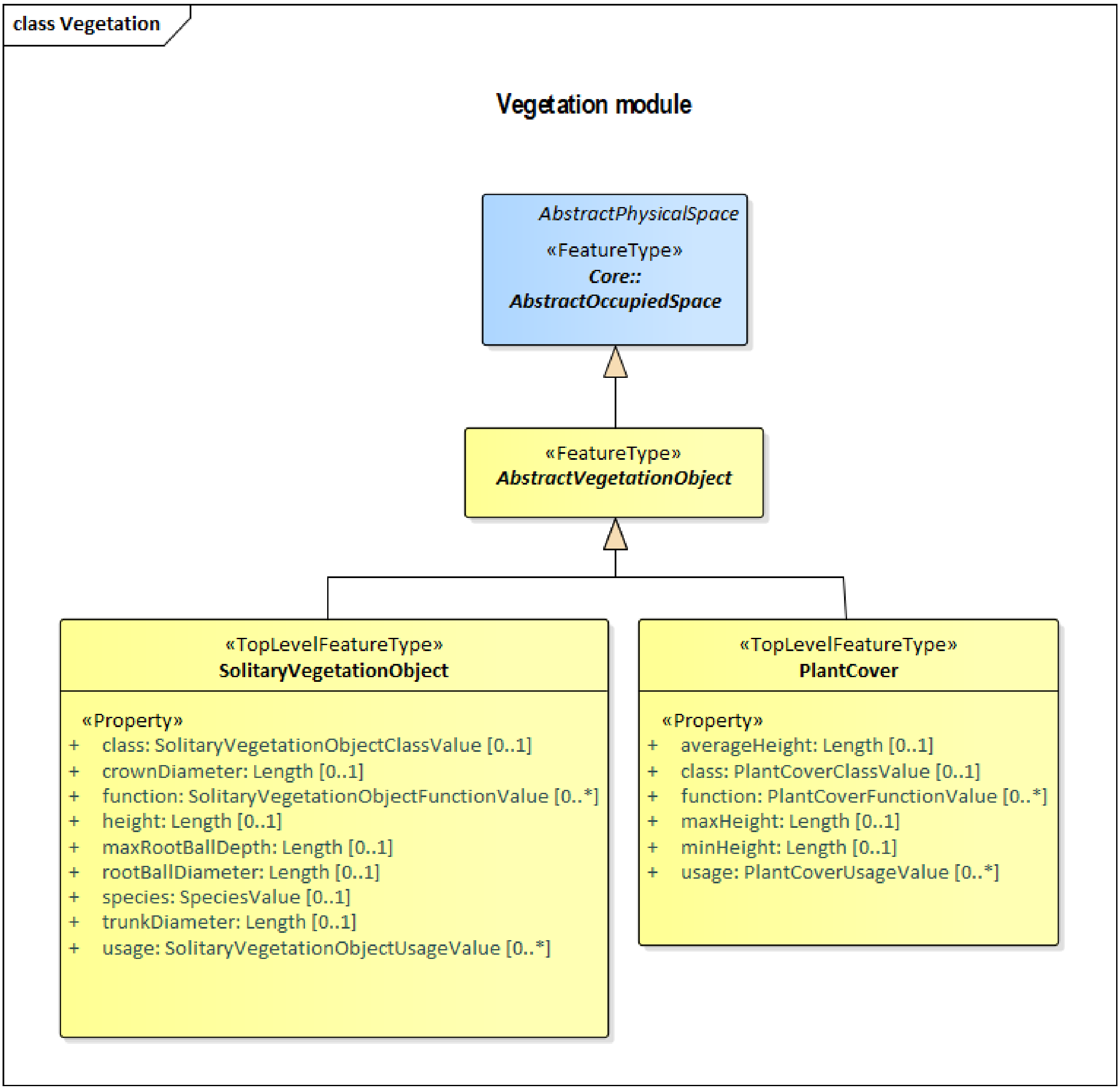


Figure 22. UML diagram of the Vegetation Model.

The UML diagram of the Vegetation Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.16. Vegetation Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.16.1. Class AbstractVegetationObject

Subclass of AbstractOccupiedSpace

Class AbstractVegetationObject	
Definition:	
Subtype Of:	AbstractOccupiedSpace
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractVegetationObject
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	PlantCover
Target Role:	
Target Class:	AbstractVegetationObject
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	SolitaryVegetationObject
Target Role:	
Target Class:	AbstractVegetationObject
Attributes	

12.16.2. Class PlantCover

Subclass of AbstractVegetationObject

Class PlantCover	
Definition:	
Subtype Of:	AbstractVegetationObject
StereoType:	«TopLevelFeatureType»
Associations	

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: PlantCover Target Role: Target Class: AbstractVegetationObject
Attributes
Attribute Name: averageHeight Value Type: Length Definition: SIG3D: Average value of the heights of the grown-up plants Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: class Value Type: PlantCoverClassValue Definition: SIG3D: Classification of PlantCover as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: PlantCoverFunctionValue Definition: SIG3D: Function of PlantCover as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: maxHeight Value Type: Length Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: minHeight Value Type: Length Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: usage Value Type: PlantCoverUsageValue Definition: SIG3D: Usage of PlantCover as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.16.3. Class PlantCoverClassValue

Subclass of <←section,>>

Class PlantCoverClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.4. Class PlantCoverFunctionValue

Subclass of <←section,>>

Class PlantCoverFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.5. Class PlantCoverUsageValue

Subclass of <←section,>>

Class PlantCoverUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.6. Class SolitaryVegetationObject

Subclass of AbstractVegetationObject

Class SolitaryVegetationObject
Definition: Subtype Of: AbstractVegetationObject StereoType: «TopLevelFeatureType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: SolitaryVegetationObject
Target Role:
Target Class: AbstractVegetationObject

Attributes

Attribute Name: class
Value Type: SolitaryVegetationObjectClassValue
Definition: SIG3D: Classification of SolitaryVegetationObject as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: crownDiameter
Value Type: Length
Definition: SIG3D: Maximal diameter of the crown.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: function
Value Type: SolitaryVegetationObjectFunctionValue
Definition: SIG3D: Function of SolitaryVegetationObject as given by the relevant national regulations, information communities, or specific applications.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: height
Value Type: Length
Definition: SIG3D: Distance between the highest point of the vegetation object and the lowest point of the terrain at the bottom of the object.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: maxRootBallDepth
Value Type: Length
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: rootBallDiameter
Value Type: Length
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: species Value Type: SpeciesValue Definition: SIG3D: Botanical classification of the SolitaryVegetationObject Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: trunkDiameter Value Type: Length Definition: SIG3D: Value of the trunk’s diameter Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: usage Value Type: SolitaryVegetationObjectUsageValue Definition: SIG3D: Usage of SolitaryVegetationObject as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.16.7. Class SolitaryVegetationObjectClassValue

Subclass of <←section,>>

Class SolitaryVegetationObjectClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.8. Class SolitaryVegetationObjectFunctionValue

Subclass of <←section,>>

Class SolitaryVegetationObjectFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.9. Class SolitaryVegetationObjectUsageValue

Subclass of <←section,>>

Class SolitaryVegetationObjectUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.10. Class SpeciesValue

Subclass of <←section,>>

Class SpeciesValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.16.11. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.16.12. Requirements

Requirement 22	**/req/vegetation/base**
A	Enter sub-requirements here

Requirement 23	**/req/vegetation/refIntegrity**
A	Enter sub-requirements here

Requirement 24	**/req/vegetation/restrictions**
A	Enter sub-requirements here

12.17. City Furniture Model

TBD

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-cityfurniture	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

The UML diagram of the City Furniture model is depicted in the City Furniture UML Diagram. The Data Dictionary for the City Furniture Package is provided in section City Furniture Data Dictionary.

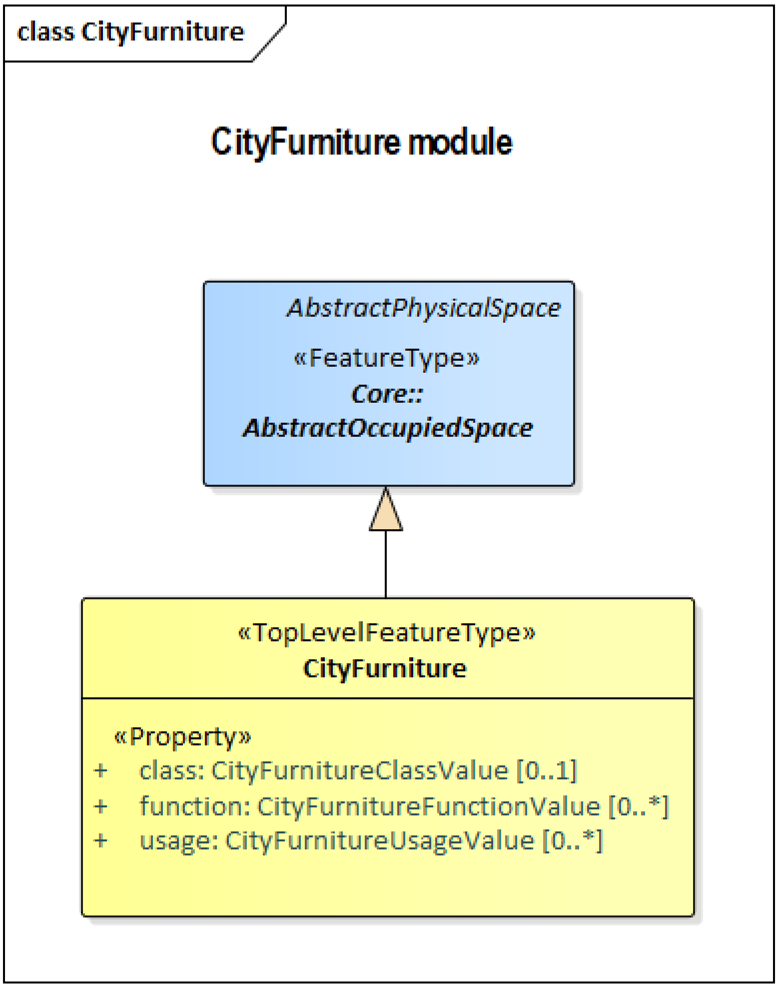


Figure 23. UML diagram of CityGML's city furniture model.

The UML diagram of CityGML's city furniture model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.18. CityFurniture Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.18.1. Class CityFurniture

Subclass of AbstractOccupiedSpace

Class CityFurniture
Definition: Subtype Of: AbstractOccupiedSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: CityFurniture Target Role: Target Class: AbstractOccupiedSpace
Attributes
Attribute Name: class Value Type: CityFurnitureClassValue Definition: Classification of CityFurniture as given by the relevant national regulations, information communities, or specific applications. [SIG3D] Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: CityFurnitureFunctionValue Definition: Specified function of CityFurniture as given by the relevant national regulations, information communities, or specific applications. [SIG3D] Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: CityFurnitureUsageValue Definition: Actual usage of CityFurniture as given by the relevant national regulations, information communities, or specific applications. [SIG3D] Multiplicity: [0..*] Stereotype: «Property»

12.18.2. Class CityFurnitureClassValue

Subclass of <←section,>>

Class CityFurnitureClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.18.3. Class CityFurnitureFunctionValue

Subclass of <←section,>>

Class CityFurnitureFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.18.4. Class CityFurnitureUsageValue

Subclass of <←section,>>

Class CityFurnitureUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.18.5. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.18.6. Requirements

Requirement 25	**/req/cityfurniture/base**
A	Enter sub-requirements here

Requirement 26	**/req/cityfurniture/refIntegrity**
A	Enter sub-requirements here

Requirement 27	**/req/cityfurniture/restrictions**
A	Enter sub-requirements here

12.19. Land Use

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-landuse	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Land Use Model is depicted in Land Use UML Diagram. The Data Dictionary for the Land Use Package is provided in section Land Use Data Dictionary.

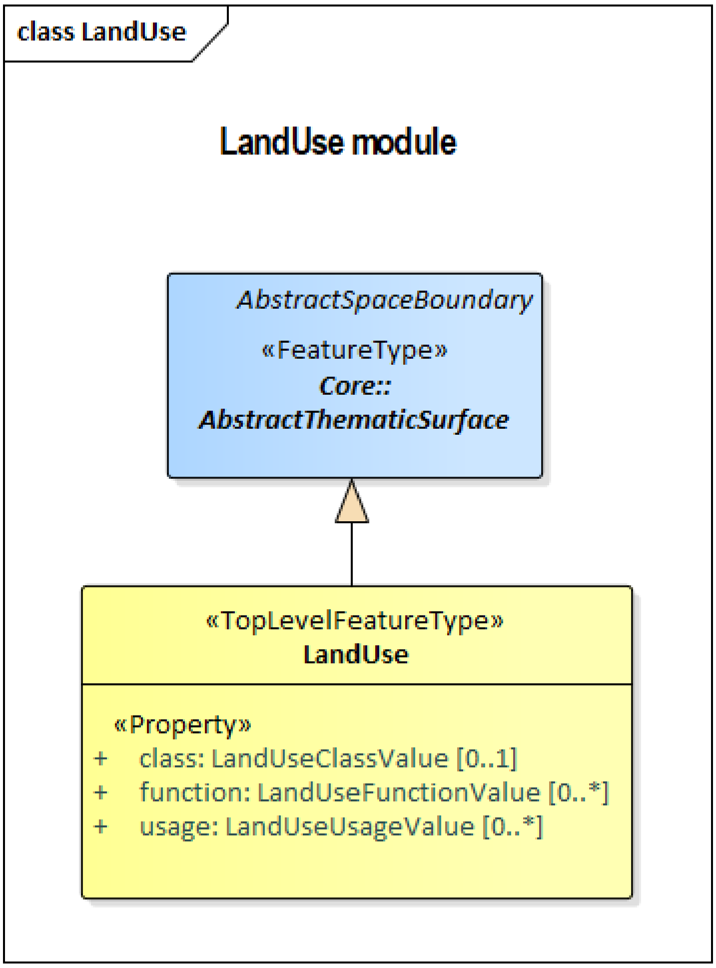


Figure 24. UML diagram of the Land Use Model.

The UML diagram of the Land Use Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.20. LandUse Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.20.1. Class LandUse

Subclass of AbstractThematicSurface

Class LandUse
Definition: Subtype Of: AbstractThematicSurface StereoType: «TopLevelFeatureType»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: LandUse Target Role: Target Class: AbstractThematicSurface
Attributes
Attribute Name: class Value Type: LandUseClassValue Definition: SIG3D: Classification of LandUse as given by the relevant national regulations, information communities or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: LandUseFunctionValue Definition: SIG3D: Specified function of LandUse as given by the relevant national regulations, information communities or specific applications. Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: LandUseUsageValue Definition: SIG3D: Actual usage of LandUse as given by the relevant national regulations, information communities or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.20.2. Class LandUseClassValue

Subclass of <←section,>>

Class LandUseClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.20.3. Class LandUseFunctionValue

Subclass of <←section,>>

Class LandUseFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.20.4. Class LandUseUsageValue

Subclass of <←section,>>

Class LandUseUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.20.5. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.20.6. Requirements

Requirement 28	**/req/landuse/base**
A	Enter sub-requirements here

Requirement 29	**/req/landuse/refIntegrity**
A	Enter sub-requirements here

Requirement 30	**/req/landuse/restrictions**
A	Enter sub-requirements here

12.21. City Object Group

Requirements Class
http://www.opengis.net/spec/CityGML/3.1/req/req-class-cityobjectgroup

Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the City Object Group Model is depicted in City Object Group UML Diagram. The Data Dictionary for the City Object Group Package is provided in section City Object Group Data Dictionary.

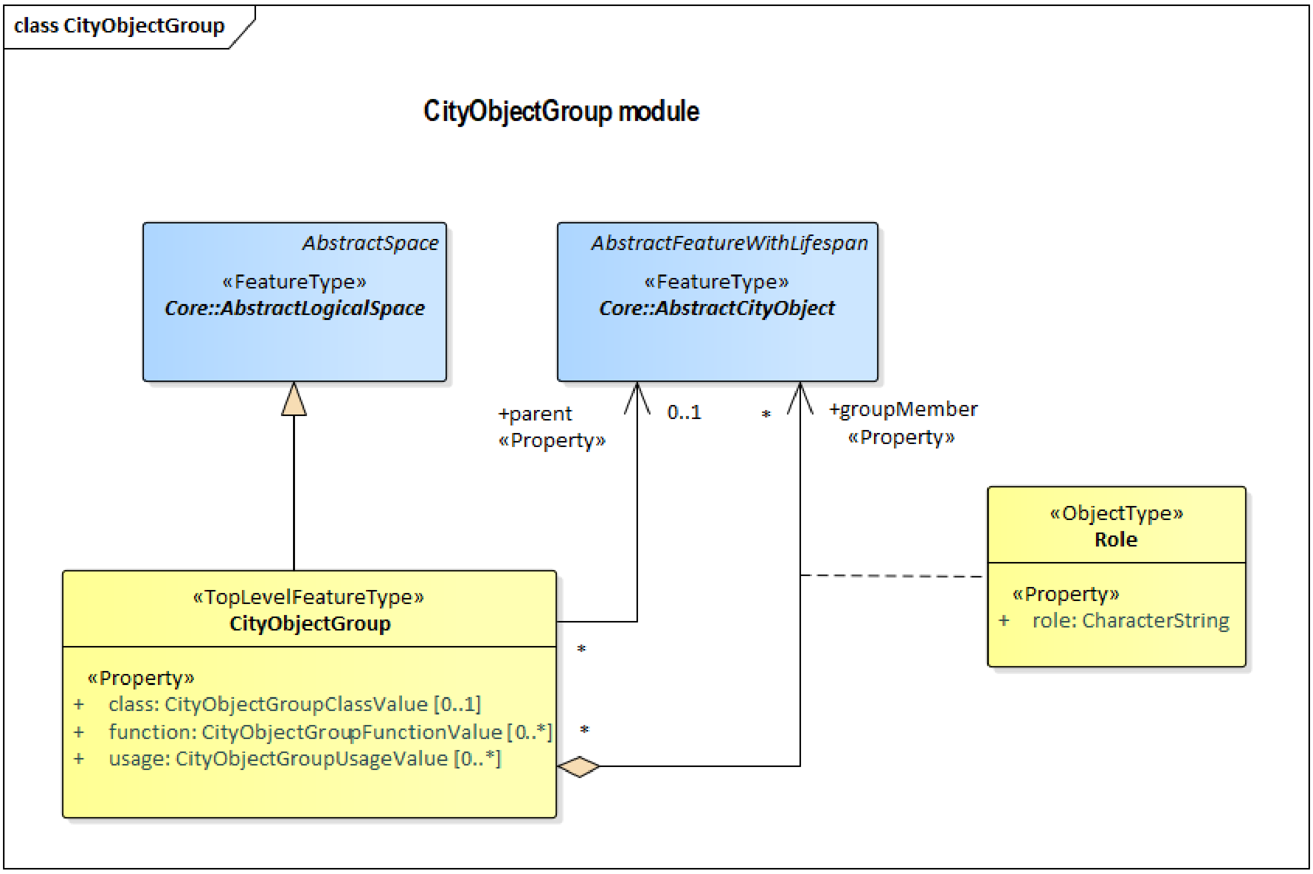


Figure 25. UML diagram of the City Object Group Model.

The UML diagram of the City Object Group Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.22. CityObjectGroup Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.22.1. Class CityObjectGroup

Subclass of AbstractLogicalSpace

Class CityObjectGroup
Definition: Subtype Of: AbstractLogicalSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: CityObjectGroup Target Role: parent Target Class: AbstractCityObject
Name: Type: AssociationClass Direction: Source → Destination Source Role: Source Class: CityObjectGroup Target Role: groupMember Target Class: AbstractCityObject
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: CityObjectGroup Target Role: Target Class: AbstractLogicalSpace
Attributes
Attribute Name: class Value Type: CityObjectGroupClassValue Definition: SIG3D: General semantical meaning of the aggregation. Classification of the aggregation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: CityObjectGroupFunctionValue Definition: SIG3D: Specific semantic meaning of the aggregation. Function of the aggregation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

Attribute Name: usage Value Type: CityObjectGroupUsageValue Definition: SIG3D: Usage of the aggregation as given by the relevant national regulations, information communities, or specific applications. Multiplicity: [0..*] Stereotype: «Property»

12.22.2. Class CityObjectGroupClassValue

Subclass of <←section,>>

Class CityObjectGroupClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.22.3. Class CityObjectGroupFunctionValue

Subclass of <←section,>>

Class CityObjectGroupFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.22.4. Class CityObjectGroupUsageValue

Subclass of <←section,>>

Class CityObjectGroupUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.22.5. Class Role

Subclass of <←section,>>

Class Role
Definition: Subtype Of: <←section,>> StereoType: «ObjectType»
Associations
Attributes
Attribute Name: role Value Type: CharacterString Definition: SIG3D: Description of the role. Multiplicity: Stereotype: «Property»

12.22.6. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.22.7. Requirements

Requirement 31	**/req/cityobjectgroup/base**
A	Enter sub-requirements here

Requirement 32	**/req/cityobjectgroup/refIntegrity**
A	Enter sub-requirements here

Requirement 33	**/req/cityobjectgroup/restrictions**
A	Enter sub-requirements here

12.23. Generics

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-generics	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Generics Model is depicted in Generics UML Diagram. The Data Dictionary for the Generics Package is provided in section Generics Data Dictionary.

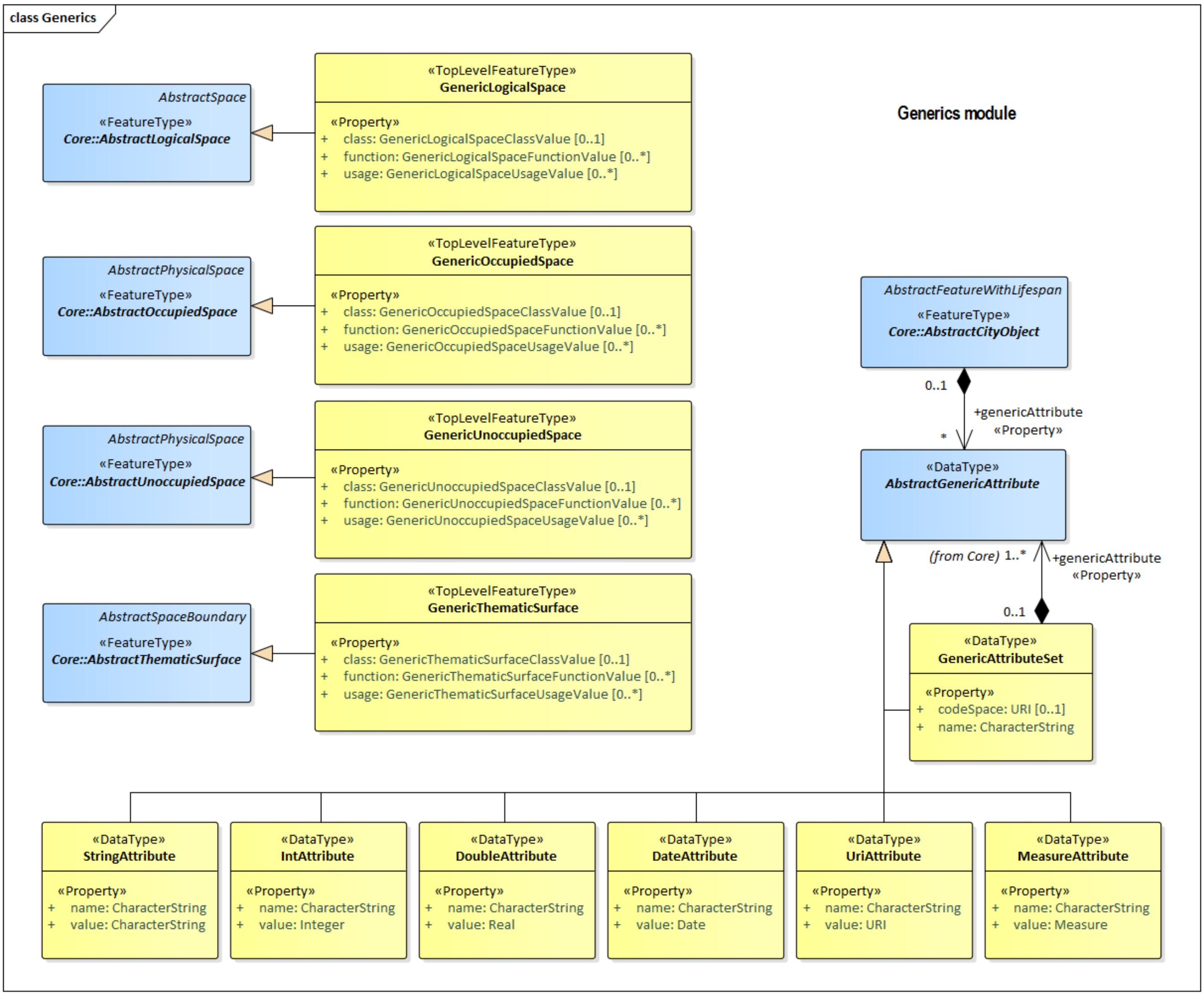


Figure 26. UML diagram of the Generics Model.

The UML diagram of the Generics Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.24. Generics Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.24.1. Class GenericLogicalSpace

Subclass of AbstractLogicalSpace

Class GenericLogicalSpace	
Definition:	
Subtype Of:	AbstractLogicalSpace
StereoType:	«TopLevelFeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GenericLogicalSpace
Target Role:	
Target Class:	AbstractLogicalSpace

Attributes	
Attribute Name:	class
Value Type:	GenericLogicalSpaceClassValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»
Attribute Name:	function
Value Type:	GenericLogicalSpaceFunctionValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»
Attribute Name:	usage
Value Type:	GenericLogicalSpaceUsageValue
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

12.24.2. Class GenericLogicalSpaceClassValue

Subclass of <←section,>>

Class GenericLogicalSpaceClassValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»

Associations

Attributes

12.24.3. Class GenericLogicalSpaceFunctionValue

Subclass of <←section,>>

Class GenericLogicalSpaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.4. Class GenericLogicalSpaceUsageValue

Subclass of <←section,>>

Class GenericLogicalSpaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.5. Class GenericOccupiedSpace

Subclass of AbstractOccupiedSpace

Class GenericOccupiedSpace
Definition: Subtype Of: AbstractOccupiedSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: GenericOccupiedSpace Target Role: Target Class: AbstractOccupiedSpace

Attributes
Attribute Name: class Value Type: GenericOccupiedSpaceClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: GenericOccupiedSpaceFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: GenericOccupiedSpaceUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.24.6. Class GenericOccupiedSpaceClassValue

Subclass of <←section,>>

Class GenericOccupiedSpaceClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.7. Class GenericOccupiedSpaceFunctionValue

Subclass of <←section,>>

Class GenericOccupiedSpaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.8. Class GenericOccupiedSpaceUsageValue

Subclass of <←section,>>

Class GenericOccupiedSpaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.9. Class GenericThematicSurface

Subclass of AbstractThematicSurface

Class GenericThematicSurface
Definition: Subtype Of: AbstractThematicSurface StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: GenericThematicSurface Target Role: Target Class: AbstractThematicSurface
Attributes
Attribute Name: class Value Type: GenericThematicSurfaceClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: GenericThematicSurfaceFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»

Attribute Name: usage Value Type: GenericThematicSurfaceUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.24.10. Class GenericThematicSurfaceClassValue

Subclass of <←section,>>

Class GenericThematicSurfaceClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.11. Class GenericThematicSurfaceFunctionValue

Subclass of <←section,>>

Class GenericThematicSurfaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.12. Class GenericThematicSurfaceUsageValue

Subclass of <←section,>>

Class GenericThematicSurfaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.13. Class GenericUnoccupiedSpace

Subclass of AbstractUnoccupiedSpace

Class GenericUnoccupiedSpace
Definition: Subtype Of: AbstractUnoccupiedSpace StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: GenericUnoccupiedSpace Target Role: Target Class: AbstractUnoccupiedSpace
Attributes
Attribute Name: class Value Type: GenericUnoccupiedSpaceClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: GenericUnoccupiedSpaceFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: GenericUnoccupiedSpaceUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.24.14. Class GenericUnoccupiedSpaceClassValue

Subclass of <←section,>>

Class GenericUnoccupiedSpaceClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations

Attributes

12.24.15. Class GenericUnoccupiedSpaceFunctionValue

Subclass of <←section,>>

Class GenericUnoccupiedSpaceFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.16. Class GenericUnoccupiedSpaceUsageValue

Subclass of <←section,>>

Class GenericUnoccupiedSpaceUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.24.17. Class DateAttribute

Subclass of AbstractGenericAttribute

Class DateAttribute
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: DateAttribute Target Role: Target Class: AbstractGenericAttribute

Attributes
Attribute Name: name Value Type: CharacterString Definition: Multiplicity: Stereotype: «Property»
Attribute Name: value Value Type: Date Definition: SIG3D: Value of the Generic Attribute. Multiplicity: Stereotype: «Property»

12.24.18. Class DoubleAttribute

Subclass of AbstractGenericAttribute

Class DoubleAttribute
Definition: Subtype Of: <←section,>> StereoType: «DataType»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: DoubleAttribute Target Role: Target Class: AbstractGenericAttribute
Attributes
Attribute Name: name Value Type: CharacterString Definition: Multiplicity: Stereotype: «Property»
Attribute Name: value Value Type: Real Definition: SIG3D: Value of the Generic Attribute. Multiplicity: Stereotype: «Property»

12.24.19. Class GenericAttributeSet

Subclass of AbstractGenericAttribute

Class GenericAttributeSet
Definition: Subtype Of: <←section,>> StereoType: «DataType»

Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: GenericAttributeSet Target Role: genericAttribute Target Class: AbstractGenericAttribute
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: GenericAttributeSet Target Role: Target Class: AbstractGenericAttribute

Attributes
Attribute Name: codeSpace Value Type: URI Definition: SIG3D: Codespace idcentifier of the Generic AttributeSet. Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: name Value Type: CharacterString Definition: Multiplicity: Stereotype: «Property»

12.24.20. Class IntAttribute

Subclass of AbstractGenericAttribute

Class IntAttribute
Definition: Subtype Of: <←section,>> StereoType: «DataType»

Associations

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	IntAttribute
Target Role:	
Target Class:	AbstractGenericAttribute
Attributes	
Attribute Name:	name
Value Type:	CharacterString
Definition:	
Multiplicity:	
Stereotype:	«Property»
Attribute Name:	value
Value Type:	Integer
Definition:	SIG3D: Value of the Generic Attribute.
Multiplicity:	
Stereotype:	«Property»

12.24.21. Class MeasureAttribute

Subclass of AbstractGenericAttribute

Class MeasureAttribute	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«DataType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	MeasureAttribute
Target Role:	
Target Class:	AbstractGenericAttribute
Attributes	
Attribute Name:	name
Value Type:	CharacterString
Definition:	
Multiplicity:	
Stereotype:	«Property»

Attribute Name: value
Value Type: Measure
Definition: SIG3D: Value of the Generic Attribute.
Multiplicity:
Stereotype: «Property»

12.24.22. Class StringAttribute

Subclass of AbstractGenericAttribute

Class StringAttribute

Definition:
Subtype Of: <←section,>>
StereoType: «DataType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: StringAttribute
Target Role:
Target Class: AbstractGenericAttribute

Attributes

Attribute Name: name
Value Type: CharacterString
Definition:
Multiplicity:
Stereotype: «Property»

Attribute Name: value
Value Type: CharacterString
Definition: SIG3D: Value of the Generic Attribute.
Multiplicity:
Stereotype: «Property»

12.24.23. Class UriAttribute

Subclass of AbstractGenericAttribute

Class UriAttribute

Definition:
Subtype Of: <←section,>>
StereoType: «DataType»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: UriAttribute Target Role: Target Class: AbstractGenericAttribute

Attributes
Attribute Name: name Value Type: CharacterString Definition: Multiplicity: Stereotype: «Property»
Attribute Name: value Value Type: URI Definition: SIG3D: Value of the Generic Attribute. Multiplicity: Stereotype: «Property»

12.24.24. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.24.25. Requirements

Requirement 34	/req/generics/base
A	Enter sub-requirements here

Requirement 35	/req/generics/refIntegrity
A	Enter sub-requirements here

Requirement 36	/req/generics/restrictions
A	Enter sub-requirements here

12.25. Construction

Requirements Class
http://www.opengis.net/spec/CityGML/3.1/req/req-class-construction

Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Construction Model is depicted in Construction UML Diagram. The Data Dictionary for the Construction Package is provided in section Construction Data Dictionary.

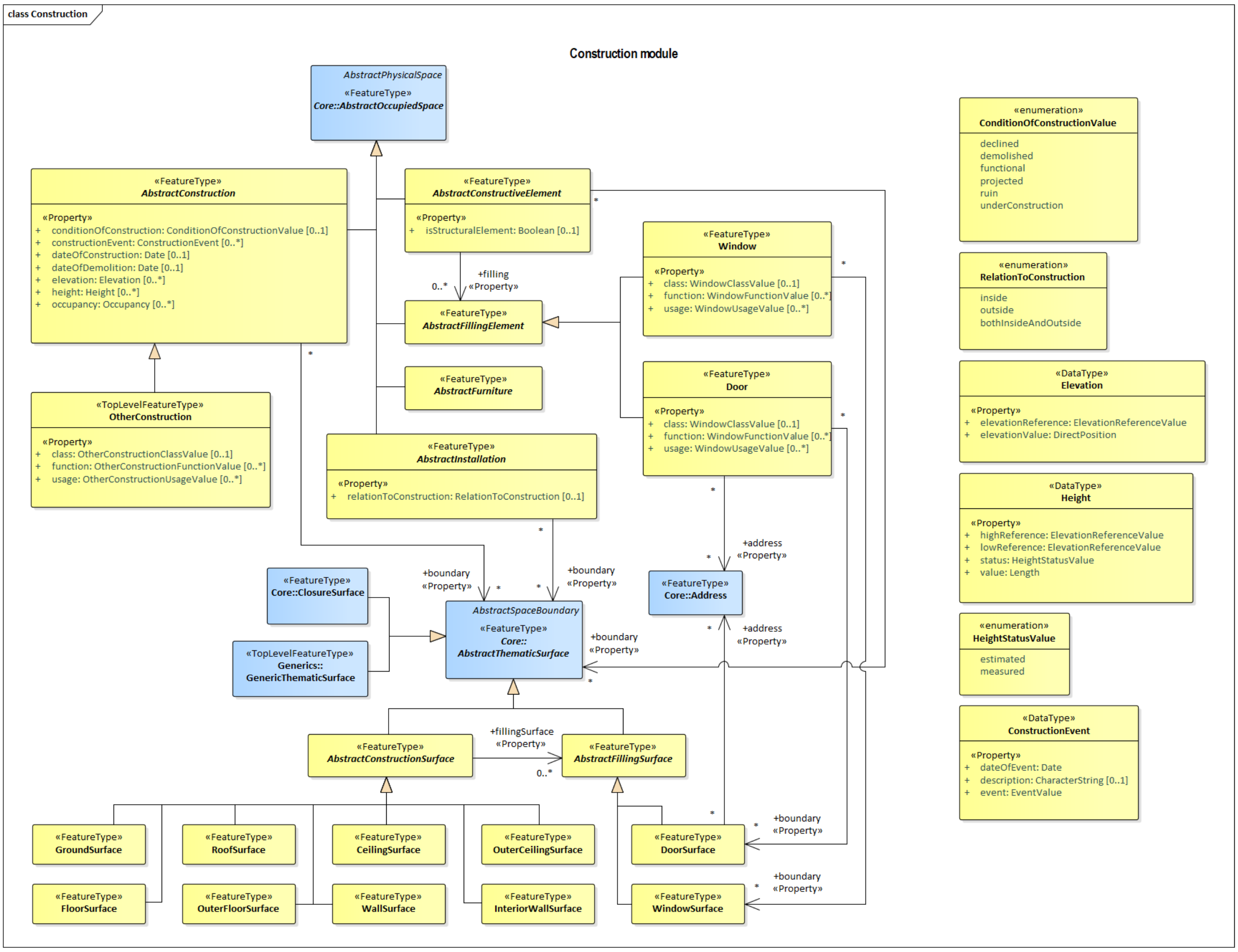


Figure 27. UML diagram of the Construction Model.

The UML diagram of the Construction Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.26. Construction Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.26.1. Class AbstractConstruction

Subclass of AbstractOccupiedSpace

Class AbstractConstruction	
Definition:	
Subtype Of:	AbstractOccupiedSpace
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstruction
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstruction
Target Role:	boundary
Target Class:	AbstractThematicSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	OtherConstruction
Target Role:	
Target Class:	AbstractConstruction
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractTunnel
Target Role:	
Target Class:	AbstractConstruction
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractBuilding
Target Role:	
Target Class:	AbstractConstruction

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: AbstractBridge
Target Role:
Target Class: AbstractConstruction

Attributes

Attribute Name: conditionOfConstruction
Value Type: ConditionOfConstructionValue
Definition: [INSPIRE] Status of the construction.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: constructionEvent
Value Type: ConstructionEvent
Definition: Date of renovation of a construction.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: dateOfConstruction
Value Type: Date
Definition: Date of completion of a construction.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: dateOfDemolition
Value Type: Date
Definition: Date of demolition of a construction.
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: elevation
Value Type: Elevation
Definition: [INSPIRE] Vertically-constrained dimensional property consisting of an absolute measure referenced to a well-defined surface which is commonly taken as origin (geoïd, water level, etc.).
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name: height
Value Type: Height
Definition: [INSPIRE] Height above ground.
Multiplicity: [0..*]
Stereotype: «Property»

Attribute Name:	occupancy
Value Type:	Occupancy
Definition:	
Multiplicity:	[0..*]
Stereotype:	«Property»

12.26.2. Class AbstractConstructionSurface

Subclass of AbstractThematicSurface

Class AbstractConstructionSurface	
Definition:	
Subtype Of:	AbstractThematicSurface
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstructionSurface
Target Role:	
Target Class:	AbstractThematicSurface
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractConstructionSurface
Target Role:	fillingSurface
Target Class:	AbstractFillingSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	FloorSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	OuterFloorSurface
Target Role:	
Target Class:	AbstractConstructionSurface

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	CeilingSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	OuterCeilingSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	RoofSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	GroundSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	InteriorWallSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	WallSurface
Target Role:	
Target Class:	AbstractConstructionSurface
Attributes	

12.26.3. Class AbstractConstructiveElement

Subclass of AbstractOccupiedSpace

Class AbstractConstructiveElement
Definition: Subtype Of: AbstractOccupiedSpace StereoType: «FeatureType»

Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractConstructiveElement Target Role: filling Target Class: AbstractFillingElement
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractConstructiveElement Target Role: boundary Target Class: AbstractThematicSurface
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractConstructiveElement Target Role: Target Class: AbstractOccupiedSpace
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BuildingConstructiveElement Target Role: Target Class: AbstractConstructiveElement
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BridgeConstructiveElement Target Role: Target Class: AbstractConstructiveElement

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelConstructiveElement Target Role: Target Class: AbstractConstructiveElement
Attributes
Attribute Name: isStructuralElement Value Type: Boolean Definition: Multiplicity: [0..1] Stereotype: «Property»

12.26.4. Class AbstractFillingElement

Subclass of AbstractOccupiedSpace

Class AbstractFillingElement
Definition: Subtype Of: AbstractOccupiedSpace StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractFillingElement Target Role: Target Class: AbstractOccupiedSpace
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractConstructiveElement Target Role: filling Target Class: AbstractFillingElement
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Window Target Role: Target Class: AbstractFillingElement

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Door Target Role: Target Class: AbstractFillingElement
Attributes

12.26.5. Class AbstractFillingSurface

Subclass of AbstractThematicSurface

Class AbstractFillingSurface
Definition: Subtype Of: AbstractThematicSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractFillingSurface Target Role: Target Class: AbstractThematicSurface
Name: Type: Association Direction: Source → Destination Source Role: Source Class: AbstractConstructionSurface Target Role: fillingSurface Target Class: AbstractFillingSurface
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WindowSurface Target Role: Target Class: AbstractFillingSurface

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: DoorSurface Target Role: Target Class: AbstractFillingSurface
Attributes

12.26.6. Class AbstractFurniture

Subclass of AbstractOccupiedSpace

Class AbstractFurniture
Definition: Subtype Of: AbstractOccupiedSpace StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: AbstractFurniture Target Role: Target Class: AbstractOccupiedSpace
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BridgeFurniture Target Role: Target Class: AbstractFurniture
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelFurniture Target Role: Target Class: AbstractFurniture

Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BuildingFurniture
Target Role:	
Target Class:	AbstractFurniture
Attributes	

12.26.7. Class AbstractInstallation

Subclass of AbstractOccupiedSpace

Class AbstractInstallation	
Definition:	
Subtype Of:	AbstractOccupiedSpace
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractInstallation
Target Role:	
Target Class:	AbstractOccupiedSpace
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	AbstractInstallation
Target Role:	boundary
Target Class:	AbstractThematicSurface
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	BridgeInstallation
Target Role:	
Target Class:	AbstractInstallation

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: TunnelInstallation Target Role: Target Class: AbstractInstallation
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: BuildingInstallation Target Role: Target Class: AbstractInstallation
Attributes
Attribute Name: relationToConstruction Value Type: RelationToConstruction Definition: Multiplicity: [0..1] Stereotype: «Property»

12.26.8. Class CeilingSurface

Subclass of AbstractConstructionSurface

Class CeilingSurface
Definition: Subtype Of: AbstractConstructionSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: CeilingSurface Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.9. Class Door

Subclass of AbstractFillingElement

Class Door
Definition: Subtype Of: AbstractFillingElement StereoType: «FeatureType»

Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Door Target Role: boundary Target Class: DoorSurface
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Door Target Role: Target Class: AbstractFillingElement
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Door Target Role: address Target Class: Address

Attributes
Attribute Name: class Value Type: WindowClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: WindowFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: WindowUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.26.10. Class DoorClassValue

Subclass of <←section,>>

Class DoorClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.11. Class DoorFunctionValue

Subclass of <←section,>>

Class DoorFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.12. Class DoorSurface

Subclass of AbstractFillingSurface

Class DoorSurface
Definition: Subtype Of: AbstractFillingSurface StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: DoorSurface Target Role: address Target Class: Address

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: DoorSurface Target Role: Target Class: AbstractFillingSurface
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Door Target Role: boundary Target Class: DoorSurface
Attributes

12.26.13. Class DoorUsageValue

Subclass of <←section,>>

Class DoorUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.14. Class ElevationReferenceValue

Subclass of <←section,>>

Class ElevationReferenceValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations

Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	ElevationReferenceValue
Attributes	

12.26.15. Class EventValue

Subclass of <←section,>>

Class EventValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»
Associations	
Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	EventValue
Attributes	

12.26.16. Class FloorSurface

Subclass of AbstractConstructionSurface

Class FloorSurface	
Definition:	
Subtype Of:	AbstractConstructionSurface
StereoType:	«FeatureType»
Associations	

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: FloorSurface Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.17. Class GroundSurface

Subclass of AbstractConstructionSurface

Class GroundSurface
Definition: Subtype Of: AbstractConstructionSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: GroundSurface Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.18. Class InteriorWallSurface

Subclass of AbstractConstructionSurface

Class InteriorWallSurface
Definition: Subtype Of: AbstractConstructionSurface StereoType: «FeatureType»
Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: InteriorWallSurface Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.19. Class OtherConstruction

Subclass of AbstractConstruction

Class OtherConstruction
Definition: Subtype Of: AbstractConstruction StereoType: «TopLevelFeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: OtherConstruction Target Role: Target Class: AbstractConstruction
Attributes
Attribute Name: class Value Type: OtherConstructionClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: OtherConstructionFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: OtherConstructionUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.26.20. Class OtherConstructionClassValue

Subclass of <←section,>>

Class OtherConstructionClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.21. Class OtherConstructionFunctionValue

Subclass of <←section,>>

Class OtherConstructionFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.22. Class OtherConstructionUsageValue

Subclass of <←section,>>

Class OtherConstructionUsageValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.23. Class OuterCeilingSurface

Subclass of AbstractConstructionSurface

Class OuterCeilingSurface
Definition: Subtype Of: AbstractConstructionSurface StereoType: «FeatureType»

Associations
Name: Type: Generalization   Direction: Source → Destination   Source Role: Source Class: OuterCeilingSurface   Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.24. Class OuterFloorSurface

Subclass of AbstractConstructionSurface

Class OuterFloorSurface
Definition: Subtype Of: AbstractConstructionSurface   StereoType: «FeatureType»

Associations
Name: Type: Generalization   Direction: Source → Destination   Source Role: Source Class: OuterFloorSurface   Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.25. Class RoofSurface

Subclass of AbstractConstructionSurface

Class RoofSurface
Definition: Subtype Of: AbstractConstructionSurface   StereoType: «FeatureType»

Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: RoofSurface Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.26. Class WallSurface

Subclass of AbstractConstructionSurface

Class WallSurface
Definition: Subtype Of: AbstractConstructionSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WallSurface Target Role: Target Class: AbstractConstructionSurface
Attributes

12.26.27. Class Window

Subclass of AbstractFillingElement

Class Window
Definition: Subtype Of: AbstractFillingElement StereoType: «FeatureType»
Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Window Target Role: Target Class: AbstractFillingElement
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Window Target Role: boundary Target Class: WindowSurface
Attributes
Attribute Name: class Value Type: WindowClassValue Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: function Value Type: WindowFunctionValue Definition: Multiplicity: [0..*] Stereotype: «Property»
Attribute Name: usage Value Type: WindowUsageValue Definition: Multiplicity: [0..*] Stereotype: «Property»

12.26.28. Class WindowClassValue

Subclass of <←section,>>

Class WindowClassValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.29. Class WindowFunctionValue

Subclass of <←section,>>

Class WindowFunctionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.30. Class WindowSurface

Subclass of AbstractFillingSurface

Class WindowSurface
Definition: Subtype Of: AbstractFillingSurface StereoType: «FeatureType»
Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: WindowSurface Target Role: Target Class: AbstractFillingSurface
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Window Target Role: boundary Target Class: WindowSurface
Attributes

12.26.31. Class WindowUsageValue

Subclass of <←section,>>

Class WindowUsageValue

Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Attributes

12.26.32. Class ConditionOfConstructionValue

Subclass of <←section,>>

Class ConditionOfConstructionValue
Definition: Subtype Of: <←section,>> StereoType:
Associations
Attributes
Attribute Name: declined Value Type: Definition: [INSPIRE] The construction cannot be used under normal conditions, though its main elements (walls, roof) are still present. Multiplicity: Stereotype:
Attribute Name: demolished Value Type: Definition: [INSPIRE] The construction has been demolished. There are no more visible remains. Multiplicity: Stereotype:
Attribute Name: functional Value Type: Definition: [INSPIRE] The construction is functional. Multiplicity: Stereotype:
Attribute Name: projected Value Type: Definition: [INSPIRE] The construction is being designed. Construction has not yet started. Multiplicity: Stereotype:

Attribute Name: ruin
Value Type:
Definition: [INSPIRE] The construction has been partly demolished and some main elements (roof, walls) have been destroyed. There are some visible remains of the construction.
Multiplicity:
Stereotype:

Attribute Name: underConstruction
Value Type:
Definition: [INSPIRE] The construction is under construction and not yet functional. This applies only to the initial construction of the construction and not to maintenance work.
Multiplicity:
Stereotype:

12.26.33. Class ConstructionEvent

Subclass of <←section,>>

Class ConstructionEvent

Definition:
Subtype Of: <←section,>>
StereoType: «DataType»

Associations

Attributes

Attribute Name: dateOfEvent
Value Type: Date
Definition:
Multiplicity:
Stereotype: «Property»

Attribute Name: description
Value Type: CharacterString
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: event
Value Type: EventValue
Definition:
Multiplicity:
Stereotype: «Property»

12.26.34. Class Elevation

Subclass of <←section,>>

Class Elevation
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Attributes
Attribute Name: elevationReference Value Type: ElevationReferenceValue Definition: [INSPIRE] Element where the elevation was measured. Multiplicity: Stereotype: «Property»
Attribute Name: elevationValue Value Type: DirectPosition Definition: [INSPIRE] Value of the elevation. Multiplicity: Stereotype: «Property»

12.26.35. Class Height

Subclass of <←section,>>

Class Height
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations
Attributes
Attribute Name: highReference Value Type: ElevationReferenceValue Definition: [INSPIRE] Element used as the high reference. Multiplicity: Stereotype: «Property»
Attribute Name: lowReference Value Type: ElevationReferenceValue Definition: [INSPIRE] Element as the low reference. Multiplicity: Stereotype: «Property»
Attribute Name: status Value Type: HeightStatusValue Definition: [INSPIRE] The way the height has been captured. Multiplicity: Stereotype: «Property»

Attribute Name:	value
Value Type:	Length
Definition:	[INSPIRE] Value of the height above ground.
Multiplicity:	
Stereotype:	«Property»

12.26.36. Class HeightStatusValue

Subclass of <←section,>>

Class HeightStatusValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	
Associations	
Attributes	
Attribute Name:	estimated
Value Type:	
Definition:	[INSPIRE] The height has been estimated and not measured.
Multiplicity:	
Stereotype:	
Attribute Name:	measured
Value Type:	
Definition:	[INSPIRE] The height has been (directly or indirectly) measured.
Multiplicity:	
Stereotype:	

12.26.37. Class RelationToConstruction

Subclass of <←section,>>

Class RelationToConstruction	
Definition:	
Subtype Of:	<←section,>>
StereoType:	
Associations	
Attributes	
Attribute Name:	inside
Value Type:	
Definition:	
Multiplicity:	
Stereotype:	

Attribute Name:	outside
Value Type:	
Definition:	
Multiplicity:	
Stereotype:	
Attribute Name:	bothInsideAndOutside
Value Type:	
Definition:	
Multiplicity:	
Stereotype:	

12.26.38. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.26.39. Requirements

Requirement 37	**/req/construction/base**
A	Enter sub-requirements here

Requirement 38	**/req/construction/refIntegrity**
A	Enter sub-requirements here

Requirement 39	**/req/construction/restrictions**
A	Enter sub-requirements here

12.27. Dynamizer

Requirements Class	
http://www.opengis.net/spec/CityGML/3.1/req/req-class-dynamizer	
Target type	Conceptual Model
Dependency	TBD
Dependency	TBD

TBD

The UML diagram of the Dynamizer Model is depicted in Dynamizer UML Diagram. The Data Dictionary for the Dynamizer Package is provided in section Dynamizer Data Dictionary.

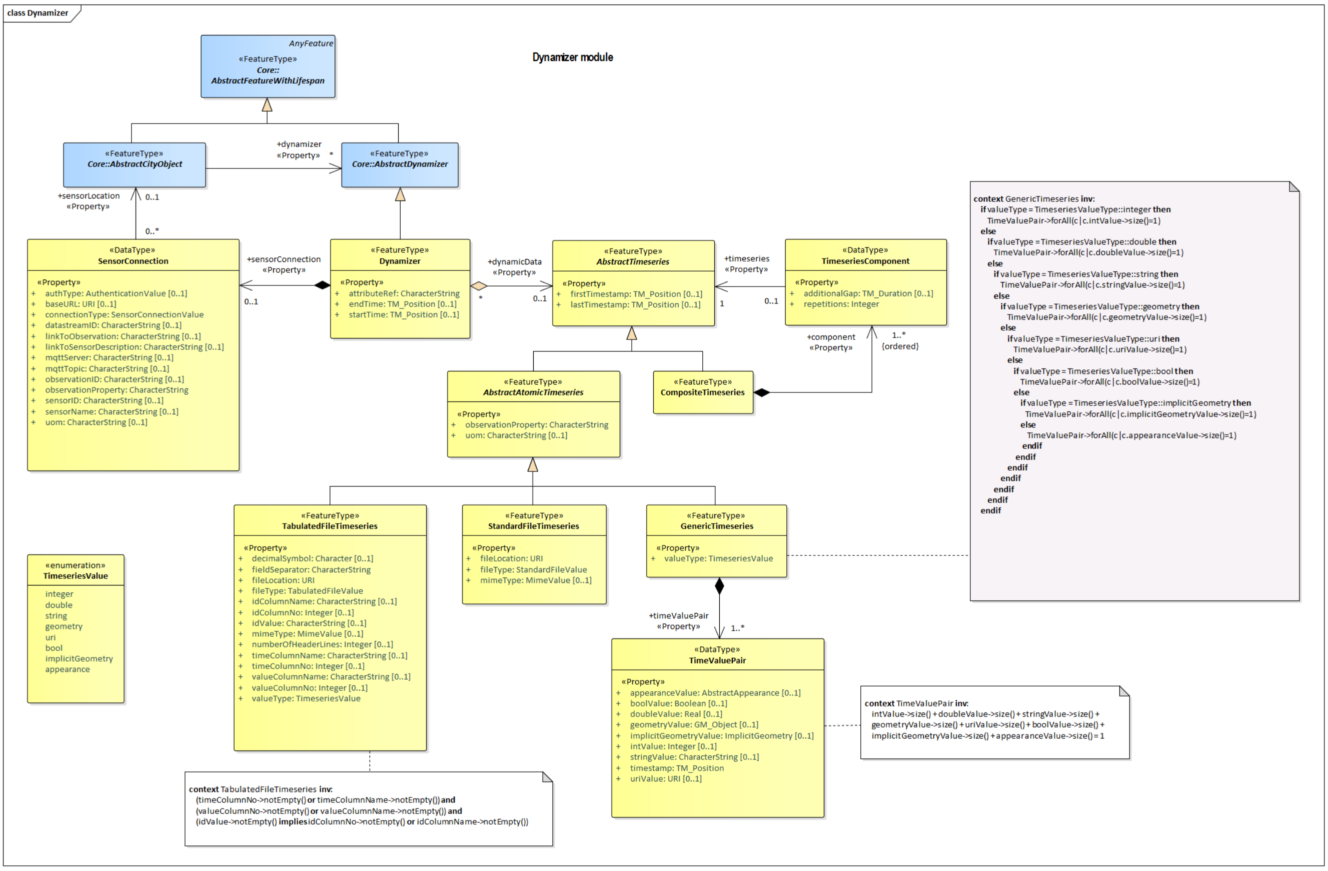


Figure 28. UML diagram of the Dynamizer Model.

The UML diagram of the Dynamizer Model. is color coded as follows:

Yellow	indicates
Blue	indicates
Pink	indicates

12.28. Dynamizer Model Data Dictionary

Description:
Stereotype: «ApplicationSchema»
Parent Package: CityGML

12.28.1. Class AbstractAtomicTimeseries

Subclass of AbstractTimeseries

Class AbstractAtomicTimeseries
Definition: Subtype Of: AbstractTimeseries StereoType: «FeatureType»
Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: AbstractAtomicTimeseries
Target Role:
Target Class: AbstractTimeseries

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: StandardFileTimeseries
Target Role:
Target Class: AbstractAtomicTimeseries

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: GenericTimeseries
Target Role:
Target Class: AbstractAtomicTimeseries

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: TabulatedFileTimeseries
Target Role:
Target Class: AbstractAtomicTimeseries

Attributes

Attribute Name: observationProperty
Value Type: CharacterString
Definition:
Multiplicity:
Stereotype: «Property»

Attribute Name: uom
Value Type: CharacterString
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

12.28.2. Class AbstractTimeseries

Subclass of <←section,>>

Class AbstractTimeseries

Definition:
Subtype Of: <←section,>>
StereoType: «FeatureType»

Associations

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: AbstractAtomicTimeseries
Target Role:
Target Class: AbstractTimeseries

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: TimeseriesComponent
Target Role: timeseries
Target Class: AbstractTimeseries

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: Dynamizer
Target Role: dynamicData
Target Class: AbstractTimeseries

Name:
Type: Generalization
Direction: Source → Destination
Source Role:
Source Class: CompositeTimeseries
Target Role:
Target Class: AbstractTimeseries

Attributes

Attribute Name: firstTimestamp
Value Type: TM_Position
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name:	lastTimestamp
Value Type:	TM_Position
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

12.28.3. Class AuthenticationValue

Subclass of <←section,>>

Class AuthenticationValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«CodeList»
Associations	
Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	AuthenticationValue
Attributes	

12.28.4. Class CompositeTimeseries

Subclass of AbstractTimeseries

Class CompositeTimeseries	
Definition:	
Subtype Of:	AbstractTimeseries
StereoType:	«FeatureType»
Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CompositeTimeseries
Target Role:	component
Target Class:	TimeseriesComponent

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: CompositeTimeseries Target Role: Target Class: AbstractTimeseries
Attributes

12.28.5. Class Dynamizer

Subclass of AbstractDynamizer

Class Dynamizer
Definition: Subtype Of: AbstractDynamizer StereoType: «FeatureType»
Associations
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Dynamizer Target Role: dynamicData Target Class: AbstractTimeseries
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: Dynamizer Target Role: Target Class: AbstractDynamizer
Name: Type: Association Direction: Source → Destination Source Role: Source Class: Dynamizer Target Role: sensorConnection Target Class: SensorConnection
Attributes

Attribute Name: attributeRef Value Type: CharacterString Definition: Multiplicity: Stereotype: «Property»
Attribute Name: endTime Value Type: TM_Position Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: startTime Value Type: TM_Position Definition: Multiplicity: [0..1] Stereotype: «Property»

12.28.6. Class GenericTimeseries

Subclass of AbstractAtomicTimeseries

Class GenericTimeseries
Definition: Subtype Of: AbstractAtomicTimeseries StereoType: «FeatureType»

Associations
Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: GenericTimeseries Target Role: Target Class: AbstractAtomicTimeseries
Name: Type: Association Direction: Source → Destination Source Role: Source Class: GenericTimeseries Target Role: timeValuePair Target Class: TimeValuePair

Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: GenericTimeseries
Attributes
Attribute Name: valueType Value Type: TimeseriesValue Definition: Multiplicity: Stereotype: «Property»

12.28.7. Class SensorConnectionValue

Subclass of <←section,>>

Class SensorConnectionValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: SensorConnectionValue
Attributes

12.28.8. Class StandardFileTimeseries

Subclass of AbstractAtomicTimeseries

Class StandardFileTimeseries
Definition: Subtype Of: AbstractAtomicTimeseries StereoType: «FeatureType»
Associations

Name: Type: Generalization Direction: Source → Destination Source Role: Source Class: StandardFileTimeseries Target Role: Target Class: AbstractAtomicTimeseries
Attributes
Attribute Name: fileLocation Value Type: URI Definition: Multiplicity: Stereotype: «Property»
Attribute Name: fileType Value Type: StandardFileValue Definition: Multiplicity: Stereotype: «Property»
Attribute Name: mimeType Value Type: MimeValue Definition: Multiplicity: [0..1] Stereotype: «Property»

12.28.9. Class StandardFileValue

Subclass of <←section,>>

Class StandardFileValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: StandardFileValue
Attributes

12.28.10. Class TabulatedFileTimeseries

Subclass of AbstractAtomicTimeseries

Class TabulatedFileTimeseries	
Definition:	
Subtype Of:	AbstractAtomicTimeseries
StereoType:	«FeatureType»

Associations	
Name:	
Type:	Generalization
Direction:	Source → Destination
Source Role:	
Source Class:	TabulatedFileTimeseries
Target Role:	
Target Class:	AbstractAtomicTimeseries
Name:	
Type:	NoteLink
Direction:	Source → Destination
Source Role:	
Source Class:	Note
Target Role:	
Target Class:	TabulatedFileTimeseries

Attributes	
Attribute Name:	decimalSymbol
Value Type:	Character
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»
Attribute Name:	fieldSeparator
Value Type:	CharacterString
Definition:	
Multiplicity:	
Stereotype:	«Property»
Attribute Name:	fileLocation
Value Type:	URI
Definition:	
Multiplicity:	
Stereotype:	«Property»
Attribute Name:	fileType
Value Type:	TabulatedFileValue
Definition:	
Multiplicity:	
Stereotype:	«Property»

Attribute Name:	idColumnName
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	idColumnNo
Value Type:	Integer
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	idValue
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	mimeType
Value Type:	MimeValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	numberOfHeaderLines
Value Type:	Integer
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	timeColumnName
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	timeColumnNo
Value Type:	Integer
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	valueColumnName
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name: valueColumnNo Value Type: Integer Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: valueType Value Type: TimeseriesValue Definition: Multiplicity: Stereotype: «Property»

12.28.11. Class TabulatedFileValue

Subclass of <←section,>>

Class TabulatedFileValue
Definition: Subtype Of: <←section,>> StereoType: «CodeList»
Associations
Name: Type: NoteLink Direction: Source → Destination Source Role: Source Class: Note Target Role: Target Class: TabulatedFileValue
Attributes

12.28.12. Class SensorConnection

Subclass of <←section,>>

Class SensorConnection
Definition: Subtype Of: <←section,>> StereoType: «DataType»
Associations

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	SensorConnection
Target Role:	sensorLocation
Target Class:	AbstractCityObject

Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	Dynamizer
Target Role:	sensorConnection
Target Class:	SensorConnection

Attributes

Attribute Name:	authType
Value Type:	AuthenticationValue
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	baseURL
Value Type:	URI
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	connectionType
Value Type:	SensorConnectionValue
Definition:	
Multiplicity:	
Stereotype:	«Property»

Attribute Name:	datastreamID
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	linkToObservation
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name: linkToSensorDescription Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: mqttServer Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: mqttTopic Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: observationID Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: observationProperty Value Type: CharacterString Definition: Multiplicity: Stereotype: «Property»
Attribute Name: sensorID Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: sensorName Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»
Attribute Name: uom Value Type: CharacterString Definition: Multiplicity: [0..1] Stereotype: «Property»

12.28.13. Class TimeseriesComponent

Subclass of <←section,>>

Class TimeseriesComponent	
Definition:	
Subtype Of:	<←section,>>
StereoType:	«DataType»

Associations	
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	TimeseriesComponent
Target Role:	timeseries
Target Class:	AbstractTimeseries
Name:	
Type:	Association
Direction:	Source → Destination
Source Role:	
Source Class:	CompositeTimeseries
Target Role:	component
Target Class:	TimeseriesComponent

Attributes	
Attribute Name:	additionalGap
Value Type:	TM_Duration
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»
Attribute Name:	repetitions
Value Type:	Integer
Definition:	
Multiplicity:	
Stereotype:	«Property»

12.28.14. Class TimeseriesValue

Subclass of <←section,>>

Class TimeseriesValue	
Definition:	
Subtype Of:	<←section,>>
StereoType:	

Associations

Attributes

Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	integer
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	double
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	string
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	geometry
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	uri
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	bool
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	implicitGeometry
Attribute Name: Value Type: Definition: Multiplicity: Stereotype:	appearance

12.28.15. Class TimeValuePair

Subclass of <←section,>>

Class TimeValuePair

Definition:
Subtype Of: <←section,>>
StereoType: «DataType»

Associations

Name:
Type: NoteLink
Direction: Source → Destination
Source Role:
Source Class: Note
Target Role:
Target Class: TimeValuePair

Name:
Type: Association
Direction: Source → Destination
Source Role:
Source Class: GenericTimeseries
Target Role: timeValuePair
Target Class: TimeValuePair

Attributes

Attribute Name: appearanceValue
Value Type: AbstractAppearance
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: boolValue
Value Type: Boolean
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: doubleValue
Value Type: Real
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name: geometryValue
Value Type: GM_Object
Definition:
Multiplicity: [0..1]
Stereotype: «Property»

Attribute Name:	implicitGeometryValue
Value Type:	ImplicitGeometry
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	intValue
Value Type:	Integer
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	stringValue
Value Type:	CharacterString
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

Attribute Name:	timestamp
Value Type:	TM_Position
Definition:	
Multiplicity:	
Stereotype:	«Property»

Attribute Name:	uriValue
Value Type:	URI
Definition:	
Multiplicity:	[0..1]
Stereotype:	«Property»

12.28.16. Additional Information

The following sections provide additional information which may not be readily available through the UML Model.

A detailed discussion of this Requirements Class can be found in the CityGML Best Practices document here.

12.28.17. Requirements

Requirement 40	**/req/dynamizer/base**
A	Enter sub-requirements here

Requirement 41	**/req/dynamizer/refIntegrity**
A	Enter sub-requirements here

Requirement 42	**/req/dynamizer/restrictions**
A	Enter sub-requirements here

Chapter 13. Media Types for any data encoding(s)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in http://www.iana.org/assignments/media-types/index.html then this section may be used to define a new MIME type for registration with IANA.

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: Title ({Normative/Informative})

NOTE Place other Annex material in sequential annexes beginning with "B" and leave final two annexes for the Revision History and Bibliography

Annex C: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version

Chapter 14. Changelog for CityGML 3.0

The following table lists all feature types, properties, and data types which have been added or changed for CityGML 3.0.

Feature Class / Data Type	Property	New	Changed	Deleted	Description of Change

Annex D: Bibliography

NOTE

Example Bibliography (Delete this note).

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

[1] OGC: OGC Testbed 12 Annex B: Architecture. (2015).