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

Copyright notice
Copyright © 2022 Open Geospatial Consortium
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/

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

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

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

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

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

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

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

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

1. Scope ........................................................................................................ 8
2. Conformance .................................................................................................. 9
   2.1. Implementation Specifications ................................................................. 9
   2.2. Conformance Classes .............................................................................. 9
3. References ..................................................................................................... 10
4. Terms and Definitions .................................................................................. 11
5. Conventions .................................................................................................. 14
   5.1. Identifiers ................................................................................................. 14
   5.2. UML Notation ........................................................................................ 14
   5.3. XML Notation ........................................................................................ 14
6. Requirements ................................................................................................ 15
   6.1. Global Requirements ............................................................................ 15
   6.2. Core ........................................................................................................ 30
   6.3. Appearance ............................................................................................. 35
   6.4. Bridge ...................................................................................................... 38
   6.5. Building ................................................................................................... 43
   6.6. CityFurniture .......................................................................................... 47
   6.7. CityObjectGroup ..................................................................................... 49
   6.8. Construction ............................................................................................ 51
   6.9. Dynamizer ............................................................................................... 56
   6.10. Generics ................................................................................................. 58
   6.11. LandUse ................................................................................................. 61
   6.12. PointCloud ............................................................................................. 62
   6.13. Relief ...................................................................................................... 64
   6.14. Transportation ....................................................................................... 65
   6.15. Tunnel ..................................................................................................... 70
   6.16. Vegetation .............................................................................................. 74
   6.17. Versioning .............................................................................................. 76
   6.18. WaterBody ............................................................................................. 78
7. Media Types .................................................................................................... 81
Annex A: Conformance Class Abstract Test Suite (Normative) ............................. 82
   A.1. Conformance Class A ............................................................................ 82
Annex B: Examples (Informative) ...................................................................... 83
   B.1. Dynamizer Examples ............................................................................ 83
   B.2. Point Cloud Examples .......................................................................... 104
   B.3. Versioning Examples ............................................................................ 108
   B.4. Address Examples ................................................................................ 116
   B.5. Application Domain Extension (ADE) Examples .................................. 118
<table>
<thead>
<tr>
<th>Annex C: XML Schema (Normative)</th>
<th>138</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1. Core</td>
<td>138</td>
</tr>
<tr>
<td>C.2. Appearance</td>
<td>170</td>
</tr>
<tr>
<td>C.3. Bridge</td>
<td>183</td>
</tr>
<tr>
<td>C.4. Building</td>
<td>194</td>
</tr>
<tr>
<td>C.5. City Furniture</td>
<td>212</td>
</tr>
<tr>
<td>C.6. City Object Group</td>
<td>213</td>
</tr>
<tr>
<td>C.7. Construction</td>
<td>216</td>
</tr>
<tr>
<td>C.8. Dynamizer</td>
<td>241</td>
</tr>
<tr>
<td>C.9. Generics</td>
<td>258</td>
</tr>
<tr>
<td>C.10. Land Use</td>
<td>268</td>
</tr>
<tr>
<td>C.11. Point Cloud</td>
<td>270</td>
</tr>
<tr>
<td>C.12. Relief</td>
<td>272</td>
</tr>
<tr>
<td>C.13. Transportation</td>
<td>279</td>
</tr>
<tr>
<td>C.14. Tunnel</td>
<td>304</td>
</tr>
<tr>
<td>C.15. Vegetation</td>
<td>315</td>
</tr>
<tr>
<td>C.16. Versioning</td>
<td>320</td>
</tr>
<tr>
<td>C.17. Water Body</td>
<td>326</td>
</tr>
</tbody>
</table>

Annex D: Conceptual Model Conformance (Normative) ................................................. 332
Annex E: Codelists (Informative) .................................................................................. 333
  E.1. Logical Model ........................................................................................................ 333
  E.2. Examples ................................................................................................................ 334
Annex F: Use of GML 3.3 (Informative) ........................................................................ 341
  F.1. Using GML 3.3 concepts in a CityGML instance document .................................... 341
  F.2. Assigning textures to compact surface geometries ............................................ 342
Annex G: Implementation Decisions (Informative) .......................................................... 344
  G.1. Encoding of association classes .......................................................................... 344
Annex H: Revision History .............................................................................................. 350
Annex I: Glossary ............................................................................................................. 351
  I.1. Abbreviated Terms ............................................................................................... 352
Annex J: Bibliography ...................................................................................................... 354
i. Abstract

The CityGML 3.0 GML Encoding Standard presents the implementation-dependent GML encoding of the concepts defined by the CityGML 3.0 Conceptual Model (CM) standard. This GML encoding can be used to store data and to exchange files via web services using the GML version 3.2.1 and 3.3. The concepts include the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical, and appearance properties. “City” is broadly defined to comprise not just built structures, but also elevation, vegetation, water bodies, city furniture, and more. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties.

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

Table 1. Conceptual Model Mapping

<table>
<thead>
<tr>
<th>Conceptual Model</th>
<th>Section</th>
<th>GML Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appearance</td>
<td>Section 6.3</td>
<td>appearance.xsd</td>
</tr>
<tr>
<td>Bridge</td>
<td>Section 6.4</td>
<td>bridge.xsd</td>
</tr>
<tr>
<td>Building</td>
<td>Section 6.5</td>
<td>building.xsd</td>
</tr>
<tr>
<td>CityFurniture</td>
<td>Section 6.6</td>
<td>cityFurniture.xsd</td>
</tr>
<tr>
<td>CityObjectGroup</td>
<td>Section 6.7</td>
<td>cityObjectGroup.xsd</td>
</tr>
<tr>
<td>Construction</td>
<td>Section 6.8</td>
<td>construction.xsd</td>
</tr>
<tr>
<td>Core</td>
<td>Section 6.2</td>
<td>core.xsd</td>
</tr>
<tr>
<td>Dynamizer</td>
<td>Section 6.9</td>
<td>dynamizer.xsd</td>
</tr>
<tr>
<td>Generics</td>
<td>Section 6.10</td>
<td>generics.xsd</td>
</tr>
<tr>
<td>LandUse</td>
<td>Section 6.11</td>
<td>landUse.xsd</td>
</tr>
<tr>
<td>PointCloud</td>
<td>Section 6.12</td>
<td>pointCloud.xsd</td>
</tr>
<tr>
<td>Relief</td>
<td>Section 6.13</td>
<td>relief.xsd</td>
</tr>
<tr>
<td>Transportation</td>
<td>Section 6.14</td>
<td>transportation.xsd</td>
</tr>
<tr>
<td>Tunnel</td>
<td>Section 6.15</td>
<td>tunnel.xsd</td>
</tr>
<tr>
<td>Vegetation</td>
<td>Section 6.16</td>
<td>vegetation.xsd</td>
</tr>
<tr>
<td>Versioning</td>
<td>Section 6.17</td>
<td>versioning.xsd</td>
</tr>
<tr>
<td>WaterBody</td>
<td>Section 6.18</td>
<td>waterBody.xsd</td>
</tr>
</tbody>
</table>

ii. Keywords

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

ogcdoc, OGC document, CityGML, 3D city models, GML, XML
iii. Preface

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

CityGML 3.0 applies the Model-Driven Architecture approach. The CityGML 3.0 Conceptual Model was defined as a Platform Independent Model (PIM). From this PIM, various platform-specific implementations can be derived. This standard defines the platform-specific implementation for the OGC GML standards 3.2.1 and 3.3 which are based on XML. The GML schemas have been derived fully automatically from the UML Conceptual Model using the software tool ShapeChange.

The CityGML 3.0 Conceptual Model standard is available under https://docs.ogc.org/is/20-010/20-010.html. An additional Users Guide providing extended explanations and examples for the individual concepts defined in the CityGML 3.0 Conceptual Model Standard is available under https://docs.ogc.org/guides/20-066.html.

As an OGC standard, CityGML follows the OGC modular specification standard, OGC 08-131r3. To be able to verify the correctness of a CityGML implementation, requirements have been defined specifying which properties can be violated. These requirements can be checked in a formalized way. Because of the breadth of CityGML, its conceptual model was divided into separate Requirements Classes, one for each subject area such as Building, Tunnel, Transportation, Vegetation, and City Furniture. This CityGML encoding similarly is divided into Requirements Classes which are then grouped into Parts. A Part may address multiple CityGML Requirements Classes but each Requirements Class is addressed in a single part. Because Requirements Classes may depend on other Requirements Classes the reader of this CityGML Part may need to conform to Requirements Classes in other Parts as well.

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

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

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

iv. Submitting organizations

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

• Heazeltech LLC
v. Submitters

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volker Coors</td>
<td>HFT Stuttgart, Germany</td>
</tr>
<tr>
<td>Charles Heazel</td>
<td>Heazeltech LLC, USA</td>
</tr>
<tr>
<td>Thomas H. Kolbe</td>
<td>Chair of Geoinformatics, Technical University of Munich, Germany</td>
</tr>
<tr>
<td>Tatjana Kutzner</td>
<td>Chair of Geoinformatics, Technical University of Munich, Germany</td>
</tr>
<tr>
<td>Claus Nagel</td>
<td>Virtual City Systems, Germany</td>
</tr>
<tr>
<td>Carl Stephen Smyth</td>
<td>OpenSitePlan, USA</td>
</tr>
</tbody>
</table>

vi. Participants in development

In addition to the Editors of the specification the following individuals contributed to this document:

Table 2. Participants in Development

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christof Beil</td>
<td>Chair of Geoinformatics, Technical University of Munich, Germany</td>
</tr>
<tr>
<td>Kanishk Chaturvedi</td>
<td>Chair of Geoinformatics, Technical University of Munich, Germany</td>
</tr>
<tr>
<td>Dean Hintz</td>
<td>Safe Software, Canada</td>
</tr>
<tr>
<td>Peter Parslow</td>
<td>Ordnance Survey, Great Britain</td>
</tr>
</tbody>
</table>
Chapter 1. Scope

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

This standard defines the PSMs and schemas for the CityGML 3.0 Implementation Specification (IS) for Geography Markup Language (GML) implementations. The GML schemas are explained in an overview and design decisions that have been made are documented as well.
Chapter 2. Conformance

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

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

2.1. Implementation Specifications

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

2.2. Conformance Classes

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

Of these seventeen conformance classes, only the Core conformance class is mandatory. All other conformance classes are optional. In the case where a conformance class has a dependency on another conformance class, that conformance class should also be implemented.
Chapter 3. References

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

- IETF: RFC 2045 & 2046, Multipurpose Internet Mail Extensions (MIME). (November 1996)
- ISO: ISO 19111:2019, Geographic information – Referencing by coordinates
- Khronos Group Inc.: COLLADA – Digital Asset Schema Release 1.5.0
- OASIS: Customer Information Quality Specifications - extensible Address Language (xAL), Version v3.0


Chapter 4. Terms and Definitions

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

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

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

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

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

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

**city-object relation**
a specific relation from the city object in which the relation is included to another city object
[CityGML Conceptual Model]

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

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

**conceptual schema**

1. formal description of a conceptual model
   [ISO 19101-1:2014, 4.1.6]

2. base schema. Formal description of the model of any geospatial information. Application
schemas are built from conceptual schemas.

OGC Definitions Register at http://www.opengis.net/def/glossary/term/ConceptualSchema

**feature**
abstraction of real world phenomena

NOTE Features are objects that have an identity that allows for distinguishing features from each other. Features can have spatial and non-spatial properties that describe the features in more detail. Spatial properties are properties that have a geometry from ISO 19107 as data type.

NOTE Features are denoted by the stereotype «FeatureType» in the CityGML 3.0 Conceptual Model.

[ISO 19101-1:2014, 4.1.11]

**geometry**
an ordered set of n-dimensional points in a given coordinate reference system; can be used to model the spatial extent or shape of a feature

NOTE Geometries are objects that can have an identity that allows for distinguishing geometries from each other. Geometries define the data types of spatial properties of features.

[OGC 15-107, modified]

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

**levels of detail**
quantity of information that portrays the real world

NOTE The concept comprises data capturing rules of spatial object types, the accuracy and the types of geometries, and other aspects of a data specification. In particular, it is related to the notions of scale and resolution.

[INSPIRE Glossary]

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

[INSPIRE Glossary]

**Platform (Model Driven Architecture)**
the set of resources on which a system is realized.

[Object Management Group, Model Driven Architecture Guide rev. 2.0]

**Platform Independent Model**
a model that is independent of a specific platform

[Object Management Group, Model Driven Architecture Guide rev. 2.0]

**Platform Specific Model**
a model of a system that is defined in terms of a specific platform

[Object Management Group, Model Driven Architecture Guide rev. 2.0]

**space**
an entity of volumetric extent in the real world

NOTE Spaces can be spatially represented in different LODs and by different types of geometric objects.

NOTE Spaces are represented in the CityGML 3.0 Conceptual Model by those classes that are derived
from the class AbstractSpace.

[CityGML Conceptual Model]

**space boundary**
an entity with areal extent in the real world. Space boundaries are objects that bound a Space. They also realize the contact between adjacent spaces.
NOTE Space boundaries can be spatially represented in different LODs and by different types of geometric objects.
NOTE Space boundaries are represented in the CityGML 3.0 Conceptual Model by those classes that are derived from the class AbstractSpaceBoundary.

[CityGML Conceptual Model]

**top-level feature**
a feature that represents one of the main components of 3D city models; can be further semantically and spatially decomposed and substructured into parts
NOTE Top-level features are denoted by the stereotype «TopLevelFeatureType» in the CityGML 3.0 Conceptual Model.

[CityGML Conceptual Model]
Chapter 5. Conventions

5.1. Identifiers

The normative provisions in this document are denoted by the URI

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

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

5.2. UML Notation

This standard is an implementation of the CityGML Conceptual Model (CM) Standard. The CityGML conceptual model was constructed using the Unified Modeling Language (UML). The UML diagrams defined in the CityGML CM standard are not repeated here; instead, the reader is directed to Chapter 7 of the CityGML CM standard.

This standard provides several UML class diagrams for illustration purposes. The UML notations used are described in the CityGML CM standard in Chapter 5.2.

In addition, this standard provides several UML object diagrams as illustrations for CityGML instance documents. In order to enhance the readability of the UML object diagrams, objects are depicted in different colors. Objects painted in orange and yellow represent features as defined in the CityGML CM standard. The orange-colored objects represent the top-most features in the feature hierarchy of the CityGML instance document, the yellow-colored objects their subordinate features. Objects painted in green represent geometries as defined in ISO 19107:2003.

5.3. XML Notation

The CityGML XML schemas derived from the CityGML CM follow the rules in OGC 07-036, GML, Annex E, UML-to-GML application schema encoding rules.
Chapter 6. Requirements

6.1. Global Requirements

This chapter defines general requirements which are valid for all Conformance Classes and cannot be derived from the GML schemas.

<table>
<thead>
<tr>
<th>Requirements Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-global">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-global</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>-</td>
</tr>
</tbody>
</table>

6.1.1. Use of GML 3.2

<table>
<thead>
<tr>
<th>Requirement 1</th>
<th>/req/global/gml</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML instance documents claiming conformance to this specification shall validate with respect to the XML schema files.</td>
<td></td>
</tr>
</tbody>
</table>

Note: This means that they have to be valid GML 3.2 documents. They may use certain aspects of CityGML 3.3; see Appendix F for further information.

The CityGML 3.0 GML schemas have been derived based on the OGC GML 3.2 standard. This means that CityGML GML instance documents must be created and exchanged in the GML version 3.2.

If for certain reasons the use of GML 3.3 is preferred, Appendix F provides an example for how to make use of GML 3.3 in CityGML instance documents. Please note, however, that software compliant to this standard might not be able to read the additional GML 3.3 content.

6.1.2. Object identifiers for features and geometries

In the GML encoding of the CityGML 3.0 Conceptual Model, different types of objects exist: features, top-level features, and geometries. All these objects have a unique identity, thus, they can be distinguished from each other and can also be referenced from other objects.

The CityGML 3.0 Conceptual Model introduces two attributes to provide all features and top-level features with a unique identity: 1) the mandatory featureID attribute to distinguish all (top-level) features and possible multiple versions of the same real-world object and 2) an optional identifier attribute to reference specific (top-level) features independent from their actual feature version. The featureID attribute value is unique within the same CityGML dataset and also for a specific version of a feature, whereas the identifier attribute has an identical value for all versions of the same real-world object. It is recommended to use globally unique identifiers like UUID values or identifiers maintained by an organization such as a mapping agency for both attributes: for the identifier attribute, as these identifiers should remain stable over the lifetime of the real-world object, and for the featureID attribute, as this ensures that 3D city models which are integrated from different sources into e.g. one GIS or database will not have colliding featureID values and also to be able to uniquely identify a specific version of a feature using these identifiers together.
with a timestamp and a version number.

The two attributes are defined in the UML class AbstractFeature as part of the Core module of the CityGML 3.0 Conceptual Model. They are mapped to the predefined GML concepts gml:id and gml:identifier in the GML encoding. Table 3 lists the two identity attributes together with their definition and GML encoding, and the GML code in Listing 1 illustrates the use of the identity attributes to represent the object identity in GML instance documents. Please note that, although the featureID attribute is defined mandatory in the CityGML 3.0 Conceptual Model, the gml:id attribute is optional according to the GML standard.

Table 3. Identity attributes used in CityGML 3.

<table>
<thead>
<tr>
<th>UML attributes to provide (top-level) features with identity</th>
<th>Mapping to corresponding GML concepts</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>featureID</td>
<td>XML attribute gml:id</td>
<td>Specifies the unique identifier of the feature that is valid in the instance document within which it occurs.</td>
</tr>
<tr>
<td>identifier</td>
<td>XML element gml:identifier</td>
<td>Specifies the unique identifier of the feature that is valid globally.</td>
</tr>
</tbody>
</table>

Listing 1. Use of the identity attributes in CityGML 3 instance documents.

```xml
<bldg:Building gml:id="B1020">
  <gml:identifier codeSpace="www.xyz.org">B1020</gml:identifier>
  <core:creationDate>2012-08-02T00:00:00</core:creationDate>
  <core:terminationDate>2013-10-09T00:00:00</core:terminationDate>
  <bldg:function>Office</bldg:function>
</bldg:Building>
```

The GML encoding implements the geometry model defined in ISO 19107. This means that all geometries are derived from the class AbstractGML as well and, thus, also geometries exhibit the same unique identity as features do. The fact that geometries also have an identity is of particular importance for CityGML, since CityGML requires defining references to geometries. For these references, only the gml:id attribute is of interest. One example is the application of textures to surfaces. The textures reference the gml:ids of those LinearRings that describe the boundaries of the surface geometries (e.g. Triangle, Polygon, or a MultiSurface consisting of Polygons) to which the textures are applied. Another example is that the CityGML GML encoding allows for realising topological relationships amongst others through referencing of shared geometries.

6.1.3. Referencing features and geometries

There are two XML concepts that are used to reference objects in GML instance documents: the XML Linking Language (XLink) and the XML Pointer Language (XPointer).

XLink allows for creating links (referred to as XLinks in this document) by providing specific attributes to XML elements for referencing other resources. The most commonly used attribute is xlink:href, the value of the attribute is the URI of the referenced document. In CityGML, however,
we are mainly interested in referencing specific objects (features and geometries) within the same or an external GML instance document. To be able to reference specific objects, XPointer needs to be used in addition to XLink. For references to objects in external documents the URI value provided in the xlink:href attribute is complemented by a number sign (#) followed by the gml:id of the object to be referenced. For references to objects within the same document it is enough to provide the number sign (#) and the gml:id of the referenced object (this is also called a relative URI). This is exemplified in Listing 2, where the element <versionMember> references a <Building> feature. Since the feature is part of the same GML instance document, a relative URI is specified by simply providing the number sign and the gml:id of the building as value of the attribute xlink:href. In the context of version management, referencing features by their gml:id means that a specific feature version of a real-world object is referenced.

Listing 2. Referencing a feature by its gml:id.

```xml
<core:CityModel gml:id="CM_1">
  <core:versionMember>
    <vers:Version gml:id="V_1">
      <vers:versionMember xlink:href="#BU_234"/>
    </vers:Version>
  </core:versionMember>
  <core:cityObjectMember>
    <bldg:Building gml:id="BU_234"> ... </bldg:Building>
  </core:cityObjectMember>
</core:CityModel>
```

When using references, it is important to keep in mind that many (top-level) features in CityGML have a complex structure. This is shown in Listing 3. In general, features can have spatial and non-spatial properties that describe the features in more detail. Spatial properties are properties that have a geometry from ISO 19107 as data type (<lod2Solid> in the example). Non-spatial properties can either have a simple data type (<function> with Integer value or <creationDate> with Date value), but they can also be further specified by additional attributes, such as the height of the building that is described by further properties (<height> with the additional properties <highReference>, <lowReference>, <status>, and <value>). In addition, they can also be composed of subfeatures that contain spatial and non-spatial properties themselves (<WallSurface> with <lod2MultiSurface> geometry).

The subfeatures can be provided inline or by reference. Inline means that the subfeatures are provided directly as content of the (top-level) feature as is shown below, where the subfeature <BuildingRoom> is provided as content of the <Building> feature. In contrast, by reference means that the subfeatures are provided elsewhere in the CityGML document and are referenced from the (top-level) feature using an XLink as is described above and illustrated in Listing 2.
6.1.4. Rules for linking features and geometries that are shared by multiple (top-level) features

When modelling cities, geometries and features can be integral parts of multiple city objects. To avoid redundant modelling of these geometries and features, CityGML offers the possibility to represent geometries and features only once and to reference them from any other city object to which they belong as well. This non-redundant representation guarantees that no integrity problems occur, i.e., several differing instances of the same geometry or feature will not exist.

Three different cases for non-redundant representation need to be differentiated:

1. Geometries are represented in different parts within the same top-level feature. An example is the roof surface of a building. The polygon representing the geometry of the RoofSurface feature is at the same time part of the RoofSurface feature and of the Solid geometry of the Building feature.

2. One geometry can be part of the representation of different features. An example is a road across a bridge, the road surface sharing the geometry with the roof surface of the bridge.
3. One and the same feature can belong to different aggregations. Examples are an intersection that belongs to two roads, the intersection being one and the same feature for both roads, or features that belong to a CityObjectGroup and that are already integral part of the city model.

For these cases, different requirements are provided for how to encode the references in CityGML. Although these requirements impose restrictions, they facilitate at the same time reading, storing, processing, and generating of CityGML documents, because they reduce the multiple possibilities of how to represent and link features and geometries in CityGML documents to the most appropriate ones. Furthermore, top-level features can now completely be loaded in the main memory, because links to shared geometries that are part of different top-level features represented further down in the GML document do not need to be resolved any more. This also facilitates querying features and geometries using web services, as up to now queries cannot address specific parts of a geometry. Maintenance at the level of the top-level features becomes easier as well, because links between feature geometries do not need to be maintained and updated any more when a feature changes its geometry or when the feature does not exist anymore.

### Requirement 2

**/req/global/referencinggeometries1**

XLinks SHALL NOT be used to reference geometries, except for geometries of ImplicitGeometry elements, from another top-level feature.

Note 1: XLinks MAY be used to reference geometries within the same top-level feature in accordance with **/req/global/referencinggeometries2** and **/req/global/referencinggeometries3**.

Note 2: ImplicitGeometry elements of different top-level features MAY reference the same geometry using XLinks. This is an exception to this requirement.

### Requirement 3

**/req/global/referencinggeometries2**

Referencing geometries of spaces and space boundaries is subject to the following restrictions:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>Geometries stored inline a space boundary SHALL NOT be redundantly stored as geometry of a space. If the geometry shall be stored with the space in addition, it SHALL reference the geometry from the space boundary using XLinks.</td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>Space boundaries SHALL NOT reference geometries of a space using XLinks.</td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>Rules A and B apply for all spaces and space boundaries that are children of the same top-level feature.</td>
</tr>
</tbody>
</table>

Note 1: The rationale of requirement A is to avoid duplicate geometries. It does NOT imply that a space has to reference all geometries of its space boundaries using XLinks. For example, the geometry can also be stored only with the space boundaries.

Note 2: If the space does not have space boundaries, then the geometry is stored with the space and no XLinks are required.

### Requirement 4

**/req/global/referencinggeometries3**

LoDs SHALL be self-contained: Geometries SHALL NOT be shared between different LoDs using XLinks.

In these requirements, XLink represents a link at the geometry level (“geometry link”), i.e., a
reference to the ID of the geometry to be reused. The link direction is always from the geometry of the space to the geometries of the space boundaries (example 1).

<table>
<thead>
<tr>
<th>Requirement 5</th>
<th>/req/global/referencinggeometries4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referencing shared geometries between top-level features using CityObjectRelations is subject to the following restrictions:</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>If two top-level features share a common geometry, the shared geometry SHALL be stored for each top-level feature separately (follows from /req/global/referencinggeometries1).</td>
</tr>
<tr>
<td>B</td>
<td>If CityObjectRelations are used to denote that a geometry is shared between two top-level features, then a CityObjectRelation SHALL be stored for each feature (might be the top-level feature itself or one of its nested features), and each CityObjectRelation SHALL be assigned the relation type “shared” and SHALL reference the other feature using an XLink. Thus, the reference SHALL be bi-directional.</td>
</tr>
</tbody>
</table>

Note: Using CityObjectRelations to link the features that share a common geometry is optional. Thus, requirement B only applies if CityObjectRelations are used.

CityObjectRelation represents a link at the feature level (“feature link”) referencing the ID of another feature that contains a shared geometry. The explicit representation of the relation between the features facilitates spatial analyses.

<table>
<thead>
<tr>
<th>Requirement 6</th>
<th>/req/global/alternativeaggregations</th>
</tr>
</thead>
<tbody>
<tr>
<td>For referencing features from alternative aggregations:</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Each feature belongs to a natural aggregation hierarchy and SHALL be stored inline this hierarchy.</td>
</tr>
<tr>
<td>B</td>
<td>Alternative aggregations SHALL NOT contain the feature inline but SHALL use an XLink to reference the feature.</td>
</tr>
</tbody>
</table>

In this requirement, XLink represents a link at the feature level (“feature link”), i.e., a reference to the ID of the feature being part of the natural aggregation. All features are part of a natural aggregation, i.e., features are typically represented in a data set once in physical form, either directly as part of the city model when they are top-level features (e.g. a Building), or inline as part of other (top-level) features (e.g. a BuildingRoom represented inline as part of the top-level feature Building). At the same time, the features can also occur in alternative aggregations.

Example 1: Building with Solid geometry and boundary surfaces
The building (=space) in Figure 1 is modelled in LOD2 as Solid geometry and is bounded by four WallSurfaces, one RoofSurface, and one GroundSurface (=space boundaries). All space boundaries are modelled as Polygon geometries. The Solid geometry of the building references the Polygon geometries using XLink. This example refers to Requirement /req/global/referencinggeometries2.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/tree/xlinks-discussion/CityGML/Examples/Building/XLink_examples/1_SimpleBuilding

The Building from the GML file is illustrated in the object diagram in Figure 2. The XLink references between the Solid geometry and the Polygon geometries are highlighted in red.
Example 2: Building with roof overhangs

The building (=space) in Figure 3 is modelled in LOD2 as Solid geometry and is bounded by four WallSurfaces, one RoofSurface, and one GroundSurface (=space boundaries). All space boundaries are modelled as Polygon geometries. The Solid geometry of the building references the Polygon geometries using XLink. This example refers to Requirement /req/global/referencinggeometries2.

The RoofSurface contains four Polygon geometries. Two of these Polygons are roof overhangs (i.e. dangling surfaces), and, thus, are not referenced by the Solid geometry of the building, as they would render the solid invalid if referenced. For this reason, an additional MultiSurface geometry is added to the building that references the dangling surfaces. In accordance with Requirement /req/global/referencinggeometries2 this MultiSurface geometry is optional. It is added to the building to provide additional information, but it is not mandatory to add this geometry.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/tree/xlinks-discussion/CityGML/Examples/Building/XLink_examples/2_SimpleBuilding_Roof_Overhangs

The Building from the GML file is illustrated in the object diagram in Figure 4. The XLink references between the Solid geometry and the Polygon geometries are highlighted in red, the XLink references between the MultiSurface geometry and the dangling surfaces in blue.
Figure 4. UML object diagram for the building in Figure 3.

Example 3: Building with BuildingInstallation

Figure 5. Building with building installation.
The building (=space) in Figure 5 is modelled in LOD2 as Solid geometry and is bounded by eight WallSurfaces, four RoofSurfaces, and one GroundSurface (=space boundaries). In addition, the building has a dormer that is modelled as BuildingInstallation (=space). The building installation is modelled as MultiSurface geometry and is bounded by one RoofSurface and three WallSurfaces (=space boundaries). This example refers to Requirement /req/global/referencinggeometries2.

The space boundaries of the building and of the building installation are all modelled as Polygon geometries. The Solid geometry of the building references those Polygon geometries that represent the space boundaries of the building space using XLink. The MultiSurface geometry of the building installation references those Polygon geometries that represent the space boundaries of the building installation using XLink. In addition, the Solid geometry may also reference the Polygon geometries that represent the space boundaries of the building installation using XLink. These references to the geometries of a nested space are optional, in accordance with Requirement /req/global/referencinggeometries2 it is also allowed to not reference these geometries.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/tree/xlinks-discussion/CityGML/Examples/Building/XLink_examples/3_Building_With_Nested_Features

The Building from the GML file is illustrated in the object diagram in Figure 6. The XLink references from the building to the space boundaries of the building are highlighted in red, whereas those to the space boundaries of the building installation are highlighted in blue.
Figure 6. UML object diagram for the building in Figure 5.

Example 4: Two buildings with shared boundary surface

Figure 7. Two buildings with shared boundary surface.
The two buildings (=top-level features) in Figure 7 are modelled in LOD2 as Solid geometry and are bounded by Wall-, Roof-, and GroundSurfaces that are modelled as Polygon geometries. One of the WallSurfaces of the first Building shares the Polygon geometry with one of the WallSurfaces of the second Building. Both WallSurfaces might appear identical, however, the surface normals of the Polygon geometries of the WallSurfaces are pointing in opposite directions. This example refers to Requirements /req/global/referencinggeometries1 and /req/global/referencinggeometries4.

To express that the WallSurfaces of the two buildings share the Polygon geometry, the WallSurfaces reference each other using a CityObjectRelation with the relation type “shared”. Both WallSurfaces contain the Polygon geometry themselves, the second WallSurface, however, in reverse order.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/tree/xlinks-discussion/CityGML/Examples/Building/XLink_examples/4_Cross-Top-Level-XLink

The Buildings from the GML file are illustrated in the object diagram in Figure 8. The CityObjectRelation is highlighted in red.

![Figure 8. UML object diagram for the building in Figure 7.](image)

**Example 5: Road crossing a Bridge**

![Figure 9. Shared surfaces between a road and a bridge.](image)
A Road and a Bridge (=top-level features) are modelled in LOD2, as is shown in Figure 9. The Bridge is bounded by Ground-, Roof-, and WallSurfaces that are modelled as MultiSurface geometries. The Road consists of three sections; each section is bounded by two TrafficAreas that are modelled as MultiSurface geometries as well. The RoofSurfaces of the Bridge share MultiSurface geometries with two TrafficAreas of the Road. The RoofSurfaces and the TrafficAreas are geometrically identical, but they differ semantically. This example refers to Requirements /req/global/referencinggeometries1 and /req/global/referencinggeometries4.

To express that the RoofSurfaces share MultiSurface geometries with two TrafficAreas, they reference each other using CityObjectRelations with the relation type “shared”.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/blob/master/CityGML/Examples/Transportation.Basic%20examples/Road_over_Bridge_CityGML3.0_LOD2_with_CityObjectRelations.gml

The Road and Bridge from the GML file are illustrated in the object diagram in Figure 10. The CityObjectRelations are highlighted in red.

Example 6: Parking garage
The parking garage in Figure 11 is modelled in LOD2 as a building (=top-level feature) with Floor-, Roof-, and WallSurfaces that are modelled as MultiSurface geometries. The parking garage contains a Road with Sections and TrafficAreas that are modelled as MultiSurface geometries as well. The Floor- and RoofSurface of the Building share MultiSurface geometries with the Sections and TrafficAreas of the Road. This example refers to Requirements /req/global/referencinggeometries1 and /req/global/referencinggeometries4.

To express the sharing of MultiSurface geometries between the Roof-/WallSurfaces and the Sections/TrafficAreas, they reference each other using CityObjectRelations with the relation type “shared”.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/blob/master/CityGML/Examples/Transportation/Basic%20examples/Road_over_Bridge_CityGML3.0_LOD2_with_CityObjectRelations.gml

Example 7: A specific version of a city model

A Version features groups, for instance, versions of city objects that are valid within a specific time period. The city model represents the natural aggregation of these versioned city objects, whereas the Version feature represents the alternative aggregation. Thus, the versioned city objects are represented inline as part of the city model, whereas they are referenced by the Version feature using XLink references. This example refers to Requirement /req/global/alternativeaggregations.

Example 8: Building rooms belonging to a Storey

BuildingRooms are usually represented inline as part of the Building they belong to, thus, the Building represents the natural aggregation. In addition, Storeys can group BuildingRooms to indicate which BuildingRoom belongs to which Storey. In this case, when the BuildingRooms are already part of the Building, the grouping into Storeys would represent an alternative aggregation, thus, the Storeys would reference the BuildingRooms using XLink relations. This example refers to Requirement /req/global/alternativeaggregations.

Example 9: A Building installation spanning across several Building Parts
Installations that are spanning across several building parts are to be physically modelled as part of one building part, all other building parts reference the installation using XLinks, expressing in this way, that the installation does not exclusively belong to one building part only. This means, that one of the BuildingParts represents the natural aggregation of the BuildingInstallation (i.e. inline representation), whereas the other BuildingParts represent alternative aggregations (i.e. XLink reference). This example refers to Requirement /req/global/alternativeaggregations.

Example 10: CityObjectGroups

A CityObjectGroup groups existing city objects that are usually represented inline somewhere else in the data set. Thus, CityObjectGroups represent alternative aggregations and have to use XLink to reference the city objects they are grouping. This example refers to Requirement /req/global/alternativeaggregations.

Exception: Intersection as part of two Roads

Figure 12. Intersection shared by two roads.

In Figure 12, two Roads (=top-level features) are shown that each have two Sections and one Intersection. The two Roads cross each other at the Intersection. Although the Intersection and, thus, also its geometry is shared by both Roads, it exists in reality only once; i.e., the Intersection is integral part of both Roads. In contrast to Requirement /req/global/referencinggeometries4, this should not be expressed by duplicating the Intersection to represent it inline of both Roads and link the duplicates using CityObjectRelations. Instead, the Intersection should be represented inline as part of one Road (here: Road 2) and be referenced by the other Road (here: Road 3) using an XLink that references the ID of the Intersection feature. This type of feature link is similar to Requirement /req/global/alternativeaggregations, where XLinks are used to relate features to alternative aggregations, with the difference that Road 3 cannot semantically be considered an alternative aggregation of the Intersection.

The GML file is available here: https://github.com/opengeospatial/CityGML-3.0Encodings/blob/master/CityGML/Examples/Transportation/Basic%20examples/ParkingGarage_CityGML3.0_LOD2_with_CityObjectRelations_and_XLinks.gml

The two Roads and the Intersection from the GML file are illustrated in the object diagram in Figure 13. The XLink reference is highlighted in red.
6.2. Core

The Core module defines the basic components of the CityGML conceptual model. This includes abstract base classes that define the core properties of more specialized thematic classes defined in other modules as well as concrete classes that are common to other modules, for example basic data types.

The concepts of the Core module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Core module is available under https://docs.ogc.org/is/20-010/20-010.html#core-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc49.

<table>
<thead>
<tr>
<th>Requirements Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>core.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

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

The applicable GML schema is core.xsd.

6.2.1. Dependencies

The Core Requirements Class is dependent on the following external OGC and ISO standard:

- GML 3.2 provides most of the geometry types (e.g., Point, LineString, Polygon) used for spatial representations in this standard, defines Coordinate Reference Systems, and supports the General Feature Model upon which this standard is based.
6.2.2. Implementation Decisions

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

1. The UML class AnyFeature is an instance of the metaclass FeatureType which is defined in ISO 19109. AnyFeature is the generalization of all feature types and acts as superclass of all classes in the CityGML UML model with the stereotype «FeatureType». A comparable concept is defined in ISO 19136 with the XML element gml:AbstractFeature as the most generic feature in GML and its basic feature model gml:AbstractFeatureType from which all feature defined in GML application schemas and, thus, also in the CityGML application schema, are derived. For this reason, the UML class AnyFeature is not encoded as CityGML-specific XML element, but is simply mapped to gml:AbstractFeature.

2. The UML class AbstractFeature is the abstract superclass of all feature types within the CityGML Conceptual Model. This class defines amongst others the attributes featureID, identifier, name, and description. These attributes represent properties that are predefined in ISO 19136 (i.e. gml:id, gml:identifier, gml:name, and gml:description) and are inherited by all features defined in GML application schemas. The attributes were added to the UML model to guarantee that they are available in all possible CityGML encodings. In the GML schema, these attributes are suppressed from being encoded, as they are already inherited by all CityGML features.

3. The UML class ImplicitGeometry is represented as an object type that defines the attribute objectID. This attribute was added to the UML model to guarantee that it is available in all possible CityGML encodings. This attribute represents the GML property gml:id which all object types automatically inherit in the GML encoding, thus, this attribute is suppressed from being encoded.

4. Several UML classes were defined in the CityGML Conceptual Model to represent concepts from the GML 3.2 standard in a language-independent way. These classes are Code, DoubleList, MeasureOrNilReasonList, DoubleOrNilReasonList, DoubleOrNilReason, NilReason, and NilReasonEnumeration. Since these concepts already exist in GML, they are simply mapped to the corresponding representations in GML. Furthermore, the class ID, representing the corresponding XML type ID, was defined as data type for the attributes featureID and objectID (see design decisions above). Since these attributes are suppressed from being encoded, also the class ID is not encoded.

5. As described in Section G.1, the UML association class CityObjectRelation is modelled as an intermediate class in an Implementation Model before the GML encoding is applied. In this way, a relation between two features A and B can be represented in such a way that feature A provides the element CityObjectRelation inline, whereas the CityObjectRelation references feature B using XLink. This encoding also ensures that Requirement /req/global/alternativeaggregations on referencing features from alternative aggregations is observed.

6. All associations allow by default that the referenced features can be provided inline or by reference. However, a restriction is defined for the following association, since it represents an alternative aggregation (see Requirement /req/global/alternativeaggregations on referencing features from alternative aggregations):

   - The reflexive association generalizesTo of the element AbstractCityObject; here, a city object must reference its generalized city object using XLink.
7. An exception to the rule that XLinks shall not be used to reference geometries from another top-level feature is defined for the element ImplicitGeometry by Requirement /req/global/referencinggeometries1. The concept of ImplicitGeometry requires that different top-level features can reference the same geometry and appearance; thus, XLinks from different top-level features to the same geometry and appearance are allowed in this case.

8. The UML class XALAddress is a placeholder for representing address details according to the OASIS xAL standard. The class is not encoded in the GML schema, but is simply mapped to xAL:Address, the root element for xAL-based address information.

9. The UML class CityModelMember was defined as a container for all the different types of objects that can occur as members of a city model. To allow that these objects can occur directly as members of a CityModel element and not as members of a CityModelMember element, this class is not mapped to a corresponding XML element, but to a global XML group.

### 6.2.3. Requirements and Recommendations

<table>
<thead>
<tr>
<th>Requirement 7</th>
<th>/req/core/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in core.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Core XML elements and their corresponding CityGML UML classes are listed in Table 4. In addition, the CityGML Core XML elements are documented in the GML schema core.xsd and in this XML Schema documentation.

#### Table 4. Core XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractAppearance</td>
<td>«FeatureType» AbstractAppearance</td>
</tr>
<tr>
<td>AbstractCityObject</td>
<td>«FeatureType» AbstractCityObject</td>
</tr>
<tr>
<td>AbstractDynamizer</td>
<td>«FeatureType» AbstractDynamizer</td>
</tr>
<tr>
<td>AbstractFeature</td>
<td>«FeatureType» AbstractFeature</td>
</tr>
<tr>
<td>AbstractFeatureWithLifespan</td>
<td>«FeatureType» AbstractFeatureWithLifespan</td>
</tr>
<tr>
<td>AbstractLogicalSpace</td>
<td>«FeatureType» AbstractLogicalSpace</td>
</tr>
<tr>
<td>AbstractOccupiedSpace</td>
<td>«FeatureType» AbstractOccupiedSpace</td>
</tr>
<tr>
<td>AbstractPhysicalSpace</td>
<td>«FeatureType» AbstractPhysicalSpace</td>
</tr>
<tr>
<td>AbstractPointCloud</td>
<td>«FeatureType» AbstractPointCloud</td>
</tr>
<tr>
<td>AbstractSpace</td>
<td>«FeatureType» AbstractSpace</td>
</tr>
<tr>
<td>AbstractSpaceBoundary</td>
<td>«FeatureType» AbstractSpaceBoundary</td>
</tr>
<tr>
<td>AbstractThematicSurface</td>
<td>«FeatureType» AbstractThematicSurface</td>
</tr>
<tr>
<td>AbstractUnoccupiedSpace</td>
<td>«FeatureType» AbstractUnoccupiedSpace</td>
</tr>
<tr>
<td>AbstractVersion</td>
<td>«FeatureType» AbstractVersion</td>
</tr>
<tr>
<td>XML Schema Element (XML Schema documentation)</td>
<td>Conceptual Model (UML model; Data Dictionary)</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>AbstractVersionTransition</td>
<td>«FeatureType» AbstractVersionTransition</td>
</tr>
<tr>
<td>Address</td>
<td>«FeatureType» Address</td>
</tr>
<tr>
<td>CityModel</td>
<td>«FeatureType» CityModel</td>
</tr>
<tr>
<td>CityObjectRelation</td>
<td>«ObjectType» CityObjectRelation</td>
</tr>
<tr>
<td>ClosureSurface</td>
<td>«FeatureType» ClosureSurface</td>
</tr>
<tr>
<td>ImplicitGeometry</td>
<td>«ObjectType» ImplicitGeometry</td>
</tr>
<tr>
<td>AbstractGenericAttribute</td>
<td>«DataType» AbstractGenericAttribute</td>
</tr>
<tr>
<td>ADEOfAbstractAppearance</td>
<td>«DataType» ADEOfAbstractAppearance</td>
</tr>
<tr>
<td>ADEOfAbstractCityObject</td>
<td>«DataType» ADEOfAbstractCityObject</td>
</tr>
<tr>
<td>ADEOfAbstractDynamizer</td>
<td>«DataType» ADEOfAbstractDynamizer</td>
</tr>
<tr>
<td>ADEOfAbstractFeature</td>
<td>«DataType» ADEOfAbstractFeature</td>
</tr>
<tr>
<td>ADEOfAbstractFeatureWithLifespan</td>
<td>«DataType» ADEOfAbstractFeatureWithLifespan</td>
</tr>
<tr>
<td>ADEOfAbstractLogicalSpace</td>
<td>«DataType» ADEOfAbstractLogicalSpace</td>
</tr>
<tr>
<td>ADEOfAbstractOccupiedSpace</td>
<td>«DataType» ADEOfAbstractOccupiedSpace</td>
</tr>
<tr>
<td>ADEOfAbstractPhysicalSpace</td>
<td>«DataType» ADEOfAbstractPhysicalSpace</td>
</tr>
<tr>
<td>ADEOfAbstractPointCloud</td>
<td>«DataType» ADEOfAbstractPointCloud</td>
</tr>
<tr>
<td>ADEOfAbstractSpace</td>
<td>«DataType» ADEOfAbstractSpace</td>
</tr>
<tr>
<td>ADEOfAbstractSpaceBoundary</td>
<td>«DataType» ADEOfAbstractSpaceBoundary</td>
</tr>
<tr>
<td>ADEOfAbstractThematicSurface</td>
<td>«DataType» ADEOfAbstractThematicSurface</td>
</tr>
<tr>
<td>ADEOfAbstractUnoccupiedSpace</td>
<td>«DataType» ADEOfAbstractUnoccupiedSpace</td>
</tr>
<tr>
<td>ADEOfAbstractVersion</td>
<td>«DataType» ADEOfAbstractVersion</td>
</tr>
<tr>
<td>ADEOfAbstractVersionTransition</td>
<td>«DataType» ADEOfAbstractVersionTransition</td>
</tr>
<tr>
<td>ADEOfAddress</td>
<td>«DataType» ADEOfAddress</td>
</tr>
<tr>
<td>ADEOfCityModel</td>
<td>«DataType» ADEOfCityModel</td>
</tr>
<tr>
<td>ADEOfClosureSurface</td>
<td>«DataType» ADEOfClosureSurface</td>
</tr>
<tr>
<td>--- (not mapped to XML, see Section 6.2.2)</td>
<td>«Union» CityModelMember</td>
</tr>
<tr>
<td>ExternalReference</td>
<td>«DataType» ExternalReference</td>
</tr>
<tr>
<td>Occupancy</td>
<td>«DataType» Occupancy</td>
</tr>
<tr>
<td>QualifiedArea</td>
<td>«DataType» QualifiedArea</td>
</tr>
<tr>
<td>QualifiedVolume</td>
<td>«DataType» QualifiedVolume</td>
</tr>
<tr>
<td>--- (not mapped to XML, see Section 6.2.2)</td>
<td>«DataType» XALAddress</td>
</tr>
</tbody>
</table>

**Requirement 8**  /req/core/reference
The following properties only allow to reference other features. The use of these properties is subject to the following restrictions:

<table>
<thead>
<tr>
<th></th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>If the <code>generalizesTo</code> property (type: <code>gml:ReferenceType</code>) of the <code>AbstractCityObject</code> element is not null, it SHALL contain an XLink reference to an <code>AbstractCityObject</code> element.</td>
</tr>
<tr>
<td>B</td>
<td>If the <code>relatedTo</code> property (type: <code>gml:ReferenceType</code>) of the <code>CityObjectRelation</code> is not null, it SHALL contain an XLink reference to an <code>AbstractCityObject</code> element.</td>
</tr>
</tbody>
</table>

Surface boundaries are constrained by the following requirement:

<table>
<thead>
<tr>
<th>Requirement 9</th>
<th>/req/core/boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table 5</strong></td>
<td>lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Core module. If a space element is bounded by thematic surface boundaries using the property <code>core:boundary</code> (type: <code>core:AbstractSpaceBoundaryPropertyType</code>), the property SHALL contain exactly one surface element from <strong>Table 5</strong> that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.</td>
</tr>
</tbody>
</table>

**Table 5** lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Core module:

**Table 5. Core space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>core:AbstractLogicalSpace</code></td>
<td>• <code>core:AbstractSpaceBoundary</code> and the substitutable elements:</td>
</tr>
<tr>
<td></td>
<td>• <code>core:AbstractThematicSurface</code>, <code>core:ClosureSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>gen:GenericThematicSurface</code></td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
<tr>
<td><code>core:AbstractOccupiedSpace</code></td>
<td>• <code>core:AbstractSpaceBoundary</code> and the substitutable elements:</td>
</tr>
<tr>
<td></td>
<td>• <code>core:AbstractThematicSurface</code>, <code>core:ClosureSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>gen:GenericThematicSurface</code></td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
</tbody>
</table>
| **core:AbstractPhysicalSpace** | • **core:AbstractSpaceBoundary** and the substitutable elements:  
  **core:AbstractThematicSurface**,  
  **core:ClosureSurface**  
  • **gen:GenericThematicSurface**  
  • possible surface elements from ADEs |
|---|---|
| **core:AbstractSpace** | • **core:AbstractSpaceBoundary** and the substitutable elements:  
  **core:AbstractThematicSurface**,  
  **core:ClosureSurface**  
  • **gen:GenericThematicSurface**  
  • possible surface elements from ADEs |
| **core:AbstractUnoccupiedSpace** | • **core:AbstractSpaceBoundary** and the substitutable elements:  
  **core:AbstractThematicSurface**,  
  **core:ClosureSurface**  
  • **gen:GenericThematicSurface**  
  • possible surface elements from ADEs |

**Recommendation 1**[/rec/core/crs]

It is *strongly recommended* that any CityGML GML instance document SHOULD explicitly specify and use a single coordinate reference system (CRS), either well-known CRS or local engineering CRS, for all contained geometry elements. This is especially important if the instance document is to be exchanged externally with third parties or is to be integrated with other spatial datasets. In addition, it simplifies the processing of the dataset by software systems.

Note: For convenience, the CRS MAY be given as value of the `srsName` attribute on the `gml:Envelope` which is the value of the `gml:boundedBy` property of the root `core:CityModel` element. Following GML rules, this CRS is inherited by all geometries contained in the instance document to any depth of nesting unless overruled by the presence of a local `srsName`.

### 6.3. Appearance

The Appearance module supports the modelling of the observable surface properties of CityGML features in the form of textures and material.

The concepts of the Appearance module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Appearance module is available under [https://docs.ogc.org/is/20-010/20-010.html#appearance-uml](https://docs.ogc.org/is/20-010/20-010.html#appearance-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc50](https://docs.ogc.org/is/20-010/20-010.html#toc50).
The Appearance Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:

- **Appearance**: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-appearance](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-appearance)

The applicable GML schema is `appearance.xsd`.

### 6.3.1. Dependencies

The Appearance Requirements Class is dependent on the following Requirements Class from this standard:

- The **Core** Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The **Core** Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.3.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference. However, a restriction is defined for the following association:
   - The association class TextureAssociation which connects ParameterizedTexture with AbstractTextureParameterization; here, AbstractTextureParameterization (or rather of the non-abstract subclasses TexCoordList and TexCoordGen) must be represented inline. For the GML encoding this means that the element ParameterizedTexture embeds the element TextureAssociation which, in turn, embeds either the element TexCoordList or TexCoordGen.

### 6.3.3. Requirements and Recommendations

<table>
<thead>
<tr>
<th>Requirement 10</th>
<th><code>/req/appearance/elements</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in <code>appearance.xsd</code>.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Appearance XML elements and their corresponding CityGML UML classes are listed in Table 6. In addition, the CityGML Appearance XML elements are documented in the GML schema `appearance.xsd` and in this XML Schema documentation.

*Table 6. Appearance XML Elements*
<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractSurfaceData</td>
<td>«FeatureType» AbstractSurfaceData</td>
</tr>
<tr>
<td>AbstractTexture</td>
<td>«FeatureType» AbstractTexture</td>
</tr>
<tr>
<td>Appearance</td>
<td>«FeatureType» Appearance</td>
</tr>
<tr>
<td>GeoreferencedTexture</td>
<td>«FeatureType» GeoreferencedTexture</td>
</tr>
<tr>
<td>ParameterizedTexture</td>
<td>«FeatureType» ParameterizedTexture</td>
</tr>
<tr>
<td>TextureAssociation</td>
<td>«ObjectType» TextureAssociation</td>
</tr>
<tr>
<td>X3DMaterial</td>
<td>«FeatureType» X3DMaterial</td>
</tr>
<tr>
<td>AbstractTextureParameterization</td>
<td>«DataType» AbstractTextureParameterization</td>
</tr>
<tr>
<td>ADEOfAbstractSurfaceData</td>
<td>«DataType» ADEOfAbstractSurfaceData</td>
</tr>
<tr>
<td>ADEOfAbstractTexture</td>
<td>«DataType» ADEOfAbstractTexture</td>
</tr>
<tr>
<td>ADEOfAppearance</td>
<td>«DataType» ADEOfAppearance</td>
</tr>
<tr>
<td>ADEOfGeoreferencedTexture</td>
<td>«DataType» ADEOfGeoreferencedTexture</td>
</tr>
<tr>
<td>ADEOfParameterizedTexture</td>
<td>«DataType» ADEOfParameterizedTexture</td>
</tr>
<tr>
<td>ADEOfX3DMaterial</td>
<td>«DataType» ADEOfX3DMaterial</td>
</tr>
<tr>
<td>TexCoordGen</td>
<td>«DataType» TexCoordGen</td>
</tr>
<tr>
<td>TexCoordList</td>
<td>«DataType» TexCoordList</td>
</tr>
</tbody>
</table>

### Requirement 11 /req/appearance/target

Surface data SHALL only be applied to surface geometries. The `target` property of a surface data element therefore SHALL only reference a subtype of `gml:AbstractSurfaceType` or a `gml:MultiSurface`.

### Requirement 12 /req/appearance/parameterizedtexture

Assigning texture coordinates to a surface geometry using ParameterizedTexture elements is subject to the following restrictions:

**A**  
Texture coordinates given by the `textureCoordinates` property of the ` TexCoordList` element define an explicit mapping of a surface's boundary points to points in texture space. A point in texture space SHALL be given as a coordinate pair consisting of two doubles.

**B**  
The `textureCoordinates` and `ring` properties of a `TexCoordList` element form pairs and their order is decisive. The first `textureCoordinates` property in the sequence forms a pair with the first `ring` property in the sequence, the second `textureCoordinates` property forms a pair with the second `ring` property, and so on. As a consequence, the number of `textureCoordinates` and `ring` properties SHALL be identical.
A TexCoordList element SHALL provide textureCoordinates for all gml:LinearRing elements contained in the surface geometry that is referenced by the target property of the embracing TextureAssociation. This explicitly includes both exterior and interior rings.

The ring property (type: anyURI) SHALL reference the gml:id of the target gml:LinearRing using an appropriate XPointer.

Each point in a ring of a surface geometry SHALL receive a point in texture space. The number of 2D points in the textureCoordinates element therefore SHALL be identical with the number of 3D points in the ring referenced by the corresponding ring property. This explicitly includes texture coordinates for the last point in a gml:LinearRing element which, by GML definition, must be coincident with the first point.

The order of points in the textureCoordinates SHALL follow the order of the points in the referenced ring element as given in the CityGML document regardless of a possibly flipped surface orientation.

Note: The order of the ring properties MAY follow the order of the gml:LinearRing elements in the target surface geometry but this is not required.

**Requirement 13**

Using GeoreferencedTexture elements is subject to the following restrictions:

A GeoreferencedTexture element SHALL define the geo-reference either inline using the referencePoint and orientation properties or externally inside the texture image (e.g., by using the GeoTIFF image format) or through an accompanying world file.

The referencePoint property (type: gml:PointPropertyType) SHALL only contain or reference a 2D point geometry.

**Recommendation 2**

The use of local appearances (i.e., appearances stored as property of a city object) SHOULD be preferred over the use of global appearances (i.e., appearances stored as appearanceMember of a CityModel).

### 6.4. Bridge

The Bridge module supports representation of thematic and spatial aspects of bridges, bridge parts, bridge installations, and interior bridge structures.

The concepts of the Bridge module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Bridge module is available under [https://docs.ogc.org/is/20-010/20-010.html#bridge-uml](https://docs.ogc.org/is/20-010/20-010.html#bridge-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc63](https://docs.ogc.org/is/20-010/20-010.html#toc63).
The Bridge Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:

- **Bridge**: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-bridge](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-bridge)

The applicable GML schema is `bridge.xsd`.

### 6.4.1. Dependencies

The Bridge Requirements Class is dependent on the following Requirements Class from this standard:

- The **Core** Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The **Core** Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.4.2. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 Bridge conformance class in GML:

1. All associations allow by default that the referenced features can be provided inline or by reference.

### 6.4.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 14</th>
<th>/req/bridge/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in <code>bridge.xsd</code>.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Bridge XML elements and their corresponding CityGML UML classes are listed in Table 7. In addition, the CityGML Bridge XML elements are documented in the GML schema `bridge.xsd` and in this XML Schema documentation.

#### Table 7. Bridge XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractBridge</td>
<td>«FeatureType» AbstractBridge</td>
</tr>
<tr>
<td>Bridge</td>
<td>«TopLevelFeatureType» Bridge</td>
</tr>
</tbody>
</table>
Surface boundaries are constrained by the following requirement:

**Requirement 15** | /req/bridge/boundaries
---|---

Table 8 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Bridge module. If a space element is bounded by thematic surface boundaries using the property `core:boundary` (type: `core:AbstractSpaceBoundaryPropertyType`), the property SHALL contain exactly one surface element from Table 8 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

Table 8 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Bridge module:

**Table 8. Bridge space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
</table>

---
| brid:AbstractBridge | • con:AbstractConstructionSurface and the substitutable elements:  
| |   con:GroundSurface,  
| |   con:RoofSurface,  
| |   con:CeilingSurface,  
| |   con:OuterCeilingSurface,  
| |   con:FloorSurface,  
| |   con:OuterFloorSurface,  
| |   con:WallSurface,  
| |   con:InteriorWallSurface  
| | • core:ClosureSurface  
| | • gen:GenericThematicSurface  
| | • possible surface elements from ADEs |
| brid:Bridge | • con:AbstractConstructionSurface and the substitutable elements:  
| |   con:GroundSurface,  
| |   con:RoofSurface,  
| |   con:CeilingSurface,  
| |   con:OuterCeilingSurface,  
| |   con:FloorSurface,  
| |   con:OuterFloorSurface,  
| |   con:WallSurface,  
| |   con:InteriorWallSurface  
| | • core:ClosureSurface  
| | • gen:GenericThematicSurface  
| | • possible surface elements from ADEs |
| brid:BridgeConstructiveElement | • con:AbstractConstructionSurface and the substitutable elements:  
| |   con:GroundSurface,  
| |   con:RoofSurface,  
| |   con:CeilingSurface,  
| |   con:OuterCeilingSurface,  
| |   con:FloorSurface,  
| |   con:OuterFloorSurface,  
| |   con:WallSurface,  
| |   con:InteriorWallSurface  
| | • core:ClosureSurface  
| | • gen:GenericThematicSurface  
| | • possible surface elements from ADEs |
| brid:BridgeFurniture | No boundaries allowed |
| **brid:BridgeInstallation** | • *con:AbstractConstructionSurface* and the substitutable elements:
  
  - *con:GroundSurface*,
  - *con:RoofSurface*,
  - *con:CeilingSurface*,
  - *con:OuterCeilingSurface*,
  - *con:FloorSurface*,
  - *con:OuterFloorSurface*,
  - *con:WallSurface*,
  - *con:InteriorWallSurface*
  
  • *core:ClosureSurface*
  
  • *gen:GenericThematicSurface*
  
  • possible surface elements from ADEs |
| **brid:BridgePart** | • *con:AbstractConstructionSurface* and the substitutable elements:
  
  - *con:GroundSurface*,
  - *con:RoofSurface*,
  - *con:CeilingSurface*,
  - *con:OuterCeilingSurface*,
  - *con:FloorSurface*,
  - *con:OuterFloorSurface*,
  - *con:WallSurface*,
  - *con:InteriorWallSurface*
  
  • *core:ClosureSurface*
  
  • *gen:GenericThematicSurface*
  
  • possible surface elements from ADEs |
| **brid:BridgeRoom** | • *con:AbstractConstructionSurface* and the substitutable elements:
  
  - *con:GroundSurface*,
  - *con:RoofSurface*,
  - *con:CeilingSurface*,
  - *con:OuterCeilingSurface*,
  - *con:FloorSurface*,
  - *con:OuterFloorSurface*,
  - *con:WallSurface*,
  - *con:InteriorWallSurface*
  
  • *core:ClosureSurface*
  
  • *gen:GenericThematicSurface*
  
  • possible surface elements from ADEs |
6.5. Building

The Building module supports representation of thematic and spatial aspects of buildings, building parts, building installations, building subdivisions, and interior building structures.

The concepts of the Building module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Building module is available under https://docs.ogc.org/is/20-010/20-010.html#building-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc64.

<table>
<thead>
<tr>
<th>Requirements Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-building">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-building</a></td>
</tr>
</tbody>
</table>

Target type  Implementation

Dependency  building.xsd

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

Dependency  http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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

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

The applicable GML schema is building.xsd.

6.5.1. Dependencies

The Building Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

6.5.2. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 Building conformance class in GML:

1. All associations allow by default that the referenced features can be provided inline or by reference.

6.5.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 16</th>
<th>/req/building/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in building.xsd.</td>
<td></td>
</tr>
</tbody>
</table>
The CityGML Building XML elements and their corresponding CityGML UML classes are listed in Table 9. In addition, the CityGML Building XML elements are documented in the GML schema building.xsd and in this XML Schema documentation.

<table>
<thead>
<tr>
<th>XML Schema Element</th>
<th>Conceptual Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Schema (XML Schema documentation)</td>
<td>(UML model; Data Dictionary)</td>
</tr>
<tr>
<td><strong>AbstractBuilding</strong></td>
<td>«FeatureType» AbstractBuilding</td>
</tr>
<tr>
<td><strong>AbstractBuildingSubdivision</strong></td>
<td>«FeatureType» AbstractBuildingSubdivision</td>
</tr>
<tr>
<td><strong>Building</strong></td>
<td>«TopLevelFeatureType» Building</td>
</tr>
<tr>
<td><strong>BuildingConstructiveElement</strong></td>
<td>«FeatureType» BuildingConstructiveElement</td>
</tr>
<tr>
<td><strong>BuildingFurniture</strong></td>
<td>«FeatureType» BuildingFurniture</td>
</tr>
<tr>
<td><strong>BuildingInstallation</strong></td>
<td>«FeatureType» BuildingInstallation</td>
</tr>
<tr>
<td><strong>BuildingPart</strong></td>
<td>«FeatureType» BuildingPart</td>
</tr>
<tr>
<td><strong>BuildingRoom</strong></td>
<td>«FeatureType» BuildingRoom</td>
</tr>
<tr>
<td><strong>BuildingUnit</strong></td>
<td>«FeatureType» BuildingUnit</td>
</tr>
<tr>
<td><strong>Storey</strong></td>
<td>«FeatureType» Storey</td>
</tr>
<tr>
<td><strong>ADEOfAbstractBuilding</strong></td>
<td>«DataType» ADEOfAbstractBuilding</td>
</tr>
<tr>
<td><strong>ADEOfAbstractBuildingSubdivision</strong></td>
<td>«DataType» ADEOfAbstractBuildingSubdivision</td>
</tr>
<tr>
<td><strong>ADEOfBuilding</strong></td>
<td>«DataType» ADEOfBuilding</td>
</tr>
<tr>
<td><strong>ADEOfBuildingConstructiveElement</strong></td>
<td>«DataType» ADEOfBuildingConstructiveElement</td>
</tr>
<tr>
<td><strong>ADEOfBuildingFurniture</strong></td>
<td>«DataType» ADEOfBuildingFurniture</td>
</tr>
<tr>
<td><strong>ADEOfBuildingInstallation</strong></td>
<td>«DataType» ADEOfBuildingInstallation</td>
</tr>
<tr>
<td><strong>ADEOfBuildingPart</strong></td>
<td>«DataType» ADEOfBuildingPart</td>
</tr>
<tr>
<td><strong>ADEOfBuildingRoom</strong></td>
<td>«DataType» ADEOfBuildingRoom</td>
</tr>
<tr>
<td><strong>ADEOfBuildingUnit</strong></td>
<td>«DataType» ADEOfBuildingUnit</td>
</tr>
<tr>
<td><strong>ADEOfStorey</strong></td>
<td>«DataType» ADEOfStorey</td>
</tr>
<tr>
<td><strong>RoomHeight</strong></td>
<td>«DataType» RoomHeight</td>
</tr>
</tbody>
</table>

Surface boundaries are constrained by the following requirement:

**Requirement 17**  | /req/building/boundaries

Table 10 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Building module. If a space element is bounded by thematic surface boundaries using the property core:boundary (type: core:AbstractSpaceBoundaryPropertyType), the property SHALL contain exactly one surface element from Table 10 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.
Table 10 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Building module:

**Table 10. Building space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bldg:AbstractBuilding</code></td>
<td>• <code>con:AbstractConstructionSurface</code> and the substitutable elements:</td>
</tr>
<tr>
<td></td>
<td>• <code>con:GroundSurface</code>, <code>con:RoofSurface</code>, <code>con:CeilingSurface</code>, <code>con:OuterCeilingSurface</code>, <code>con:FloorSurface</code>, <code>con:OuterFloorSurface</code>, <code>con:WallSurface</code>, <code>con:InteriorWallSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>core:ClosureSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>gen:GenericThematicSurface</code></td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
<tr>
<td><code>bldg:AbstractBuildingSubdivision</code></td>
<td>No boundaries allowed</td>
</tr>
<tr>
<td><code>bldg:bldg:Building</code></td>
<td>• <code>con:AbstractConstructionSurface</code> and the substitutable elements:</td>
</tr>
<tr>
<td></td>
<td>• <code>con:GroundSurface</code>, <code>con:RoofSurface</code>, <code>con:CeilingSurface</code>, <code>con:OuterCeilingSurface</code>, <code>con:FloorSurface</code>, <code>con:OuterFloorSurface</code>, <code>con:WallSurface</code>, <code>con:InteriorWallSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>core:ClosureSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>gen:GenericThematicSurface</code></td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
</tbody>
</table>
| bldg:BuildingConstructiveElement | • con:AbstractConstructionSurface and the substitutable elements:
  con:GroundSurface,
  con:RoofSurface,
  con:CeilingSurface,
  con:OuterCeilingSurface,
  con:FloorSurface,
  con:OuterFloorSurface,
  con:WallSurface,
  con:InteriorWallSurface
  • core:ClosureSurface
  • gen:GenericThematicSurface
  • possible surface elements from ADEs |
| bldg:BuildingFurniture | No boundaries allowed |
| bldg:BuildingInstallation | • con:AbstractConstructionSurface and the substitutable elements:
  con:GroundSurface,
  con:RoofSurface,
  con:CeilingSurface,
  con:OuterCeilingSurface,
  con:FloorSurface,
  con:OuterFloorSurface,
  con:WallSurface,
  con:InteriorWallSurface
  • core:ClosureSurface
  • gen:GenericThematicSurface
  • possible surface elements from ADEs |
| bldg:BuildingPart | • con:AbstractConstructionSurface and the substitutable elements:
  con:GroundSurface,
  con:RoofSurface,
  con:CeilingSurface,
  con:OuterCeilingSurface,
  con:FloorSurface,
  con:OuterFloorSurface,
  con:WallSurface,
  con:InteriorWallSurface
  • core:ClosureSurface
  • gen:GenericThematicSurface
  • possible surface elements from ADEs |
### 6.6. CityFurniture

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

The concepts of the CityFurniture module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the City Furniture module is available under [https://docs.ogc.org/is/20-010/20-010.html#cityfurniture-uml](https://docs.ogc.org/is/20-010/20-010.html#cityfurniture-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc51](https://docs.ogc.org/is/20-010/20-010.html#toc51).
The CityFurniture Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:

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

The applicable GML schema is cityFurniture.xsd.

### 6.6.1. Dependencies

The CityFurniture Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.6.2. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 CityFurniture conformance class in GML:

1. All associations allow by default that the referenced features can be provided inline or by reference.

### 6.6.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 18</th>
<th>/req/cityfurniture/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in cityFurniture.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML CityFurniture XML elements and their corresponding CityGML UML classes are listed in Table 11. In addition, the CityGML CityFurniture XML elements are documented in the GML schema cityFurniture.xsd and in this XML Schema documentation.

#### Table 11. CityFurniture XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityFurniture</td>
<td>«TopLevelFeatureType» CityFurniture</td>
</tr>
</tbody>
</table>
Surface boundaries are constrained by the following requirement:

**Requirement 19**  
/req/cityfurniture/boundaries

Table 12 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the CityFurniture module. If a space element is bounded by thematic surface boundaries using the property `core:boundary` (type: `core:AbstractSpaceBoundaryPropertyType`), the property SHALL contain exactly one surface element from Table 12 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

Table 12 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the CityFurniture module:

**Table 12. CityFurniture space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
</table>
| `frn:CityFurniture` | • `core:ClosureSurface`  
|                | • `gen:GenericThematicSurface`  
|                | • possible surface elements from ADEs                        |

### 6.7. CityObjectGroup

The CityObjectGroup module supports grouping of city objects. Arbitrary city objects may be aggregated in groups according to user-defined criteria. A group may be further classified by application-specific attributes.

The concepts of the CityObjectGroup module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the CityObjectGroup module is available under [https://docs.ogc.org/is/20-010/20-010.html#cityobject-uml](https://docs.ogc.org/is/20-010/20-010.html#cityobject-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc52](https://docs.ogc.org/is/20-010/20-010.html#toc52).

**Requirements Class**

- [http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-cityobjectgroup](http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-cityobjectgroup)
- Target type: Implementation
- Dependency: `cityObjectGroup.xsd`
- Dependency: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityobjectgroup](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityobjectgroup)
- Dependency: [http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core](http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core)

The CityObjectGroup Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:
6.7.1. Dependencies

The CityObjectGroup Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

6.7.2. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 CityObjectGroup conformance class in GML:

1. As described in Section G.1, the UML association class Role is modelled as an intermediate class in an Implementation Model before the GML encoding is applied. In this way, a relation between a city object group and the features it groups can be represented in such a way that the city object group provides the Role element inline, whereas the Role element references the features that belong to the city object group using XLink. This encoding also ensures that Requirement /req/global/alternativeaggregations on referencing features from alternative aggregations is observed.

2. All associations allow by default that the referenced features can be provided inline or by reference. However, a restriction is defined for the following association, since it represents an alternative aggregation (see Requirement /req/global/alternativeaggregations on referencing features from alternative aggregations):
   - The association parent which connects CityObjectGroup with AbstractCityObject; here, a city object group must reference its parent city object using XLink.

6.7.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 20</th>
<th>/req/cityobjectgroup/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in cityObjectGroup.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML CityObjectGroup XML elements and their corresponding CityGML UML classes are listed in Table 13. In addition, the CityGML CityObjectGroup XML elements are documented in the GML schema cityObjectGroup.xsd and in this XML Schema documentation.

Table 13. CityObjectGroup XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityObjectGroup</td>
<td>«TopLevelFeatureType» CityObjectGroup</td>
</tr>
</tbody>
</table>
### Requirement 21  
/req/cityobjectgroup/reference

The following properties only allow to reference other features. The use of these properties is subject to the following restrictions:

- **A** If the `parent` property (type: `gml:ReferenceType`) of the `CityObjectGroup` is not null, it SHALL contain an XLink reference to a `core:AbstractCityObject` element.
- **B** If the `groupMember` property (type: `gml:AbstractFeatureMemberType`) of the `Role` element is not null, it SHALL contain an XLink reference to a `core:AbstractCityObject` element.

Surface boundaries are constrained by the following requirement:

### Requirement 22  
/req/cityobjectgroup/boundaries

Table 14 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the CityObjectGroup module. If a space element is bounded by thematic surface boundaries using the property `core:boundary` (type: `core:AbstractSpaceBoundaryPropertyType`), the property SHALL contain exactly one surface element from Table 14 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

**Table 14. CityObjectGroup space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>grp:CityObjectGroup</code></td>
<td>• <code>core:ClosureSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>gen:GenericThematicSurface</code></td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
</tbody>
</table>

### 6.8. Construction

The Construction module supports representation of key elements of different types of constructions. These key elements include construction surfaces (e.g. floor and ceiling), windows and doors, constructive elements (e.g. beams and slabs), installations, and furniture.

The concepts of the Construction module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Construction module is available under [https://docs.ogc.org/is/20-010/20-010.html#construction-uml](https://docs.ogc.org/is/20-010/20-010.html#construction-uml), the Data Dictionary is available under [https://docs.ogc.org/is/](https://docs.ogc.org/is/)
### Requirements Class

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>construction.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-construction">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-construction</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

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

The applicable GML schema is `construction.xsd`.

#### 6.8.1. Dependencies

The Construction Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

#### 6.8.2. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 Construction conformance class in GML:

1. All associations allow by default that the referenced features can be provided inline or by reference.

#### 6.8.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 23</th>
<th>/req/construction/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in <code>construction.xsd</code>.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Construction XML elements and their corresponding CityGML UML classes are listed in Table 15. In addition, the CityGML Construction XML elements are documented in the GML schema `construction.xsd` and in this XML Schema documentation.

*Table 15. Construction XML Elements*
<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractConstruction</td>
<td>«FeatureType» AbstractConstruction</td>
</tr>
<tr>
<td>AbstractConstructionSurface</td>
<td>«FeatureType» AbstractConstructionSurface</td>
</tr>
<tr>
<td>AbstractConstructiveElement</td>
<td>«FeatureType» AbstractConstructiveElement</td>
</tr>
<tr>
<td>AbstractFillingElement</td>
<td>«FeatureType» AbstractFillingElement</td>
</tr>
<tr>
<td>AbstractFillingSurface</td>
<td>«FeatureType» AbstractFillingSurface</td>
</tr>
<tr>
<td>AbstractFurniture</td>
<td>«FeatureType» AbstractFurniture</td>
</tr>
<tr>
<td>AbstractInstallation</td>
<td>«FeatureType» AbstractInstallation</td>
</tr>
<tr>
<td>CeilingSurface</td>
<td>«FeatureType» CeilingSurface</td>
</tr>
<tr>
<td>Door</td>
<td>«FeatureType» Door</td>
</tr>
<tr>
<td>DoorSurface</td>
<td>«FeatureType» DoorSurface</td>
</tr>
<tr>
<td>FloorSurface</td>
<td>«FeatureType» FloorSurface</td>
</tr>
<tr>
<td>GroundSurface</td>
<td>«FeatureType» GroundSurface</td>
</tr>
<tr>
<td>InteriorWallSurface</td>
<td>«FeatureType» InteriorWallSurface</td>
</tr>
<tr>
<td>OtherConstruction</td>
<td>«TopLevelFeatureType» OtherConstruction</td>
</tr>
<tr>
<td>OuterCeilingSurface</td>
<td>«FeatureType» OuterCeilingSurface</td>
</tr>
<tr>
<td>OuterFloorSurface</td>
<td>«FeatureType» OuterFloorSurface</td>
</tr>
<tr>
<td>RoofSurface</td>
<td>«FeatureType» RoofSurface</td>
</tr>
<tr>
<td>WallSurface</td>
<td>«FeatureType» WallSurface</td>
</tr>
<tr>
<td>Window</td>
<td>«FeatureType» Window</td>
</tr>
<tr>
<td>WindowSurface</td>
<td>«FeatureType» WindowSurface</td>
</tr>
<tr>
<td>ADEOfAbstractConstruction</td>
<td>«DataType» ADEOfAbstractConstruction</td>
</tr>
<tr>
<td>ADEOfAbstractConstructionSurface</td>
<td>«DataType» ADEOfAbstractConstructionSurface</td>
</tr>
<tr>
<td>ADEOfAbstractConstructiveElement</td>
<td>«DataType» ADEOfAbstractConstructiveElement</td>
</tr>
<tr>
<td>ADEOfAbstractFillingElement</td>
<td>«DataType» ADEOfAbstractFillingElement</td>
</tr>
<tr>
<td>ADEOfAbstractFillingSurface</td>
<td>«DataType» ADEOfAbstractFillingSurface</td>
</tr>
<tr>
<td>ADEOfAbstractFurniture</td>
<td>«DataType» ADEOfAbstractFurniture</td>
</tr>
<tr>
<td>ADEOfAbstractInstallation</td>
<td>«DataType» ADEOfAbstractInstallation</td>
</tr>
<tr>
<td>ADEOfCeilingSurface</td>
<td>«DataType» ADEOfCeilingSurface</td>
</tr>
<tr>
<td>ADEOfDoor</td>
<td>«DataType» ADEOfDoor</td>
</tr>
<tr>
<td>ADEOfDoorSurface</td>
<td>«DataType» ADEOfDoorSurface</td>
</tr>
<tr>
<td>ADEOfFloorSurface</td>
<td>«DataType» ADEOfFloorSurface</td>
</tr>
<tr>
<td>ADEOfGroundSurface</td>
<td>«DataType» ADEOfGroundSurface</td>
</tr>
<tr>
<td>ADEOfInteriorWallSurface</td>
<td>«DataType» ADEOfInteriorWallSurface</td>
</tr>
<tr>
<td>ADEOfWindow</td>
<td>«DataType» ADEOfWindow</td>
</tr>
</tbody>
</table>
Surface boundaries are constrained by the following requirement:

**Requirement 24**  
/req/construction/boundaries

Table 16 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Construction module. If a space element is bounded by thematic surface boundaries using the property `core:boundary` (type: `core:AbstractSpaceBoundaryPropertyType`), the property SHALL contain exactly one surface element from Table 16 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

Table 16 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Construction module:

**Table 16. Construction space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
</table>
| `con:AbstractConstruction` | • `con:AbstractConstructionSurface` and the substitutable elements:  
  `con:GroundSurface`,  
  `con:RoofSurface`,  
  `con:CeilingSurface`,  
  `con:OuterCeilingSurface`,  
  `con:FloorSurface`,  
  `con:OuterFloorSurface`,  
  `con:WallSurface`,  
  `con:InteriorWallSurface`  
  • `core:ClosureSurface`  
  • `gen:GenericThematicSurface`  
  • possible surface elements from ADEs |
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
</table>
| `con:AbstractConstructiveElement` | - `con:AbstractConstructionSurface` and the substitutable elements:<br>  
  - `con:GroundSurface`,  
  - `con:RoofSurface`,  
  - `con:CeilingSurface`,  
  - `con:OuterCeilingSurface`,  
  - `con:FloorSurface`,  
  - `con:OuterFloorSurface`,  
  - `con:WallSurface`,  
  - `con:InteriorWallSurface`  
  - `core:ClosureSurface`  
  - `gen:GenericThematicSurface`  
  - possible surface elements from ADEs |
| `con:AbstractFillingElement`  | No boundaries allowed                                                                         |
| `con:AbstractFurniture`       | No boundaries allowed                                                                         |
| `con:AbstractInstallation`    | - `con:AbstractConstructionSurface` and the substitutable elements:<br>  
  - `con:GroundSurface`,  
  - `con:RoofSurface`,  
  - `con:CeilingSurface`,  
  - `con:OuterCeilingSurface`,  
  - `con:FloorSurface`,  
  - `con:OuterFloorSurface`,  
  - `con:WallSurface`,  
  - `con:InteriorWallSurface`  
  - `core:ClosureSurface`  
  - `gen:GenericThematicSurface`  
  - possible surface elements from ADEs |
| `con:Door`                    | - `con:DoorSurface`  
  - `core:ClosureSurface`  
  - `gen:GenericThematicSurface`  
  - possible surface elements from ADEs |
• con:AbstractConstructionSurface and the substitutable elements:
  - con:GroundSurface,
  - con:RoofSurface,
  - con:CeilingSurface,
  - con:OuterCeilingSurface,
  - con:FloorSurface,
  - con:OuterFloorSurface,
  - con:WallSurface,
  - con:InteriorWallSurface
• core:ClosureSurface
• gen:GenericThematicSurface
• possible surface elements from ADEs

• con:WindowSurface
• core:ClosureSurface
• gen:GenericThematicSurface
• possible surface elements from ADEs

### 6.9. Dynamizer

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

The concepts of the Dynamizer module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Dynamizer module is available under [https://docs.ogc.org/is/20-010/20-010.html#dynamizer-uml](https://docs.ogc.org/is/20-010/20-010.html#dynamizer-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc53](https://docs.ogc.org/is/20-010/20-010.html#toc53).

<table>
<thead>
<tr>
<th>Requirements Class</th>
<th><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-dynamizer">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-dynamizer</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Target type</td>
<td>Implementation</td>
</tr>
<tr>
<td>Dependency</td>
<td>dynamizer.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-dynamizer">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-dynamizer</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

The Dynamizer Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:
The applicable GML schema is `dynamizer.xsd`.

### 6.9.1. Dependencies

The Dynamizer Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.9.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference. However, a restriction is defined for the following association, since it represents an alternative aggregation (see Requirement `/req/global/alternativeaggregations` on referencing features from alternative aggregations):

   - The association sensorLocation which connects SensorConnection with AbstractCityObject; here, a sensor connection must reference the city object for which it provides sensor information using XLink.

### 6.9.3. Requirements

**Requirement 25** `/req/dynamizer/elements`

CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in `dynamizer.xsd`.

The CityGML Dynamizer XML elements and their corresponding CityGML UML classes are listed in Table 17. In addition, the CityGML Dynamizer XML elements are documented in the GML schema `dynamizer.xsd` and in this XML Schema documentation.

#### Table 17. Dynamizer XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractAtomicTimeseries</td>
<td>«FeatureType» AbstractAtomicTimeseries</td>
</tr>
<tr>
<td>AbstractTimeseries</td>
<td>«FeatureType» AbstractTimeseries</td>
</tr>
<tr>
<td>CompositeTimeseries</td>
<td>«FeatureType» CompositeTimeseries</td>
</tr>
<tr>
<td>Dynamizer</td>
<td>«FeatureType» Dynamizer</td>
</tr>
<tr>
<td>GenericTimeseries</td>
<td>«FeatureType» GenericTimeseries</td>
</tr>
<tr>
<td>StandardFileTimeseries</td>
<td>«FeatureType» StandardFileTimeseries</td>
</tr>
<tr>
<td>TabulatedFileTimeseries</td>
<td>«FeatureType» TabulatedFileTimeseries</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>ADEOfAbstractAtomicTimeseries</td>
<td>«DataType» ADEOfAbstractAtomicTimeseries</td>
</tr>
<tr>
<td>ADEOfAbstractTimeseries</td>
<td>«DataType» ADEOfAbstractTimeseries</td>
</tr>
<tr>
<td>ADEOfCompositeTimeseries</td>
<td>«DataType» ADEOfCompositeTimeseries</td>
</tr>
<tr>
<td>ADEOfDynamizer</td>
<td>«DataType» ADEOfDynamizer</td>
</tr>
<tr>
<td>ADEOfGenericTimeseries</td>
<td>«DataType» ADEOfGenericTimeseries</td>
</tr>
<tr>
<td>ADEOfStandardFileTimeseries</td>
<td>«DataType» ADEOfStandardFileTimeseries</td>
</tr>
<tr>
<td>ADEOfTabulatedFileTimeseries</td>
<td>«DataType» ADEOfTabulatedFileTimeseries</td>
</tr>
<tr>
<td>SensorConnection</td>
<td>«DataType» SensorConnection</td>
</tr>
<tr>
<td>TimeseriesComponent</td>
<td>«DataType» TimeseriesComponent</td>
</tr>
<tr>
<td>TimeValuePair</td>
<td>«DataType» TimeValuePair</td>
</tr>
</tbody>
</table>

### Requirement 26: /req/dynamizer/reference

If the `sensorLocation` property (type: `gml:ReferenceType`) of the `SensorConnection` element is not null, it SHALL contain an XLink reference to a `core:AbstractCityObject` element.

### Requirement 27: /req/dynamizer/attributeref

Referencing a feature property value using the `attributeRef` property of a `Dynamizer` element is subject to the following restrictions:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The <code>attributeRef</code> property (type: xs:string) SHALL reference a feature property value using XPath.</td>
</tr>
<tr>
<td>B</td>
<td>The feature property referenced by the <code>attributeRef</code> property SHALL exist.</td>
</tr>
<tr>
<td>C</td>
<td>The data type of the timeseries data provided by a <code>Dynamizer</code> element SHALL be equal to the data type of the static value of the feature property that is referenced by the <code>attributeRef</code> property or be of a type that can be converted to the type of the feature property without information loss.</td>
</tr>
</tbody>
</table>

### 6.10. Generics

The Generics module supports application-specific extensions to the CityGML conceptual model. These extensions may be used to model and exchange additional attributes and features not covered by the predefined thematic classes of CityGML. Generic extensions shall only be used if appropriate thematic classes or attributes are not provided by any other CityGML module.

The concepts of the Generics module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Generics module is available under [https://docs.ogc.org/is/20-010/20-010.html#generics-uml](https://docs.ogc.org/is/20-010/20-010.html#generics-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc54](https://docs.ogc.org/is/20-010/20-010.html#toc54).
The Generics Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:

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

The applicable GML schema is generics.xsd.

### 6.10.1. Dependencies

The Generics Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.10.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference.

### 6.10.3. Requirements

**Requirement 28** /req/generics/elements

CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in generics.xsd.

The CityGML Generics XML elements and their corresponding CityGML UML classes are listed in Table 18. In addition, the CityGML Generics XML elements are documented in the GML schema generics.xsd and in this XML Schema documentation.

**Table 18. Generics XML Elements**

<table>
<thead>
<tr>
<th>XML Schema Element</th>
<th>Conceptual Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>(XML Schema documentation)</td>
<td>(UML model; Data Dictionary)</td>
</tr>
<tr>
<td>GenericLogicalSpace</td>
<td>«TopLevelFeatureType» GenericLogicalSpace</td>
</tr>
</tbody>
</table>
Surface boundaries are constrained by the following requirement:

**Requirement 29**

Table 19 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Generics module. If a space element is bounded by thematic surface boundaries using the property `core:boundary` (type: `core:AbstractSpaceBoundaryPropertyType`), the property SHALL contain exactly one surface element from Table 19 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

**Table 19. Generics space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gen:GenericLogicalSpace</code></td>
<td>• <code>core:ClosureSurface</code></td>
</tr>
<tr>
<td></td>
<td>• <code>gen:GenericThematicSurface</code></td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
</tbody>
</table>
6.11. LandUse

The LandUse module supports representation of areas of the earth's surface dedicated to a specific land use.

The concepts of the LandUse module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the LandUse module is available under https://docs.ogc.org/is/20-010/20-010.html#landuse-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc55.

### Requirements Class

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>landUse.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-landuse">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-landuse</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

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

The applicable GML schema is landUse.xsd.

### 6.11.1. Dependencies

The LandUse Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.11.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference.

### 6.11.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 30</th>
<th>/req/landuse/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in landUse.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML LandUse XML elements and their corresponding CityGML UML classes are listed in Table 20. In addition, the CityGML LandUse XML elements are documented in the GML schema landUse.xsd and in this XML Schema documentation.

#### Table 20. LandUse XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LandUse</td>
<td>«TopLevelFeatureType» LandUse</td>
</tr>
<tr>
<td>ADEOfLandUse</td>
<td>«DataType» ADEOfLandUse</td>
</tr>
</tbody>
</table>

### 6.12. PointCloud

The PointCloud module supports representation of CityGML features by a collection of points.

The concepts of the PointCloud module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the PointCloud module is available under [https://docs.ogc.org/is/20-010/20-010.html#pointcloud-uml](https://docs.ogc.org/is/20-010/20-010.html#pointcloud-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc56](https://docs.ogc.org/is/20-010/20-010.html#toc56).

#### Requirements Class

<table>
<thead>
<tr>
<th>Requirements Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-pointcloud">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-pointcloud</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>pointCloud.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

- **PointCloud**: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud)

The applicable GML schema is pointCloud.xsd.
6.12.1. Dependencies

The PointCloud Requirements Class is dependent on the following Requirements Class from this standard:

- The **Core** Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The **Core** Requirements Class also defines XML elements and types reused by other Requirements Classes.

6.12.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference.

2. The PointCloud module allows for representing the geometries of city objects by 3D point clouds in three ways. Examples for each representation option are provided in Section B.2:
   - A MultiPoint geometry is provided inline with the CityGML file, i.e., the point cloud of a specific feature is embedded directly within the feature represented by the point cloud.
   - A separate point cloud file is provided for every feature. Each feature in the CityGML file references the corresponding point cloud file.
   - One point cloud file is provided that contains all points from all features in a specific area. In the point cloud file, all points belonging to a specific feature have the same value set in the attribute “point_source_id”. Each feature in the CityGML instance document references the point cloud file and all points with the corresponding value in the attribute “point_source_id”.

6.12.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 31</th>
<th>/req/pointcloud/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in pointCloud.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML PointCloud XML elements and their corresponding CityGML UML classes are listed in Table 21. In addition, the CityGML PointCloud XML elements are documented in the GML schema pointCloud.xsd and in this XML Schema documentation.

Table 21. PointCloud XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointCloud</td>
<td>«FeatureType» PointCloud</td>
</tr>
<tr>
<td>ADEOfPointCloud</td>
<td>«DataType» ADEOfPointCloud</td>
</tr>
</tbody>
</table>
6.13. Relief

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

The concepts of the Relief module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Relief module is available under https://docs.ogc.org/is/20-010/20-010.html#relief-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc57.

### Requirements Class

<table>
<thead>
<tr>
<th>Requirements Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-relief">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-relief</a></td>
<td></td>
</tr>
<tr>
<td>Target type</td>
<td>Implementation</td>
</tr>
<tr>
<td>Dependency</td>
<td>relief.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

- **Relief**: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief)

The applicable GML schema is relief.xsd.

### 6.13.1. Dependencies

The Relief Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.13.2. Implementation Decisions

The following decisions have been made regarding implementation of the CityGML 3.0 Relief conformance class in GML:

1. All associations allow by default that the referenced features can be provided inline or by reference.

### 6.13.3. Requirements

| Requirement 32 | /req/relief/elements |
CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in relief.xsd.

The CityGML Relief XML elements and their corresponding CityGML UML classes are listed in Table 22. In addition, the CityGML Relief XML elements are documented in the GML schema relief.xsd and in this XML Schema documentation.

Table 22. Relief XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractReliefComponent</td>
<td>«FeatureType» AbstractReliefComponent</td>
</tr>
<tr>
<td>BreaklineRelief</td>
<td>«FeatureType» BreaklineRelief</td>
</tr>
<tr>
<td>MassPointRelief</td>
<td>«FeatureType» MassPointRelief</td>
</tr>
<tr>
<td>RasterRelief</td>
<td>«FeatureType» RasterRelief</td>
</tr>
<tr>
<td>ReliefFeature</td>
<td>«TopLevelFeatureType» ReliefFeature</td>
</tr>
<tr>
<td>TINRelief</td>
<td>«FeatureType» TINRelief</td>
</tr>
<tr>
<td>ADEOfAbstractReliefComponent</td>
<td>«DataType» ADEOfAbstractReliefComponent</td>
</tr>
<tr>
<td>ADEOfBreaklineRelief</td>
<td>«DataType» ADEOfBreaklineRelief</td>
</tr>
<tr>
<td>ADEOfMassPointRelief</td>
<td>«DataType» ADEOfMassPointRelief</td>
</tr>
<tr>
<td>ADEOfRasterRelief</td>
<td>«DataType» ADEOfRasterRelief</td>
</tr>
<tr>
<td>ADEOfReliefFeature</td>
<td>«DataType» ADEOfReliefFeature</td>
</tr>
<tr>
<td>ADEOfTINRelief</td>
<td>«DataType» ADEOfTINRelief</td>
</tr>
</tbody>
</table>

6.14. Transportation

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

The concepts of the Transportation module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Transportation module is available under https://docs.ogc.org/is/20-010/20-010.html#transportation-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc58.

Requirements Class

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

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td>transportation.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-transportation">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-transportation</a></td>
</tr>
</tbody>
</table>
The Transportation Conformance Class implements the following Requirements Class from the CityGML 3.0 Conceptual Model Standard:

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

The applicable GML schema is transportation.xsd.

### 6.14.1. Dependencies

The Transportation Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.14.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference. However, restrictions are defined for the following associations, since they represent alternative aggregations (see Requirement /req/global/alternativeaggregations on referencing features from alternative aggregations):
   - The reflexive association successor of TrafficSpace; here, a traffic space must reference its succeeding traffic space using XLink.
   - The reflexive association predecessor of TrafficSpace; here, a traffic space must reference its preceding traffic space using XLink.

### 6.14.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 33</th>
<th>/req/transportation/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in transportation.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Transportation XML elements and their corresponding CityGML UML classes are listed in Table 23. In addition, the CityGML Transportation XML elements are documented in the GML schema transportation.xsd and in this XML Schema documentation.

**Table 23. Transportation XML Elements**

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractTransportationSpace</td>
<td>«FeatureType» AbstractTransportationSpace</td>
</tr>
</tbody>
</table>

66
<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuxiliaryTrafficArea</td>
<td>«FeatureType» AuxiliaryTrafficArea</td>
</tr>
<tr>
<td>AuxiliaryTrafficSpace</td>
<td>«FeatureType» AuxiliaryTrafficSpace</td>
</tr>
<tr>
<td>ClearanceSpace</td>
<td>«FeatureType» ClearanceSpace</td>
</tr>
<tr>
<td>Hole</td>
<td>«FeatureType» Hole</td>
</tr>
<tr>
<td>HoleSurface</td>
<td>«FeatureType» HoleSurface</td>
</tr>
<tr>
<td>Intersection</td>
<td>«FeatureType» Intersection</td>
</tr>
<tr>
<td>Marking</td>
<td>«FeatureType» Marking</td>
</tr>
<tr>
<td>Railway</td>
<td>«TopLevelFeatureType» Railway</td>
</tr>
<tr>
<td>Road</td>
<td>«TopLevelFeatureType» Road</td>
</tr>
<tr>
<td>Section</td>
<td>«FeatureType» Section</td>
</tr>
<tr>
<td>Square</td>
<td>«TopLevelFeatureType» Square</td>
</tr>
<tr>
<td>Track</td>
<td>«TopLevelFeatureType» Track</td>
</tr>
<tr>
<td>TrafficArea</td>
<td>«FeatureType» TrafficArea</td>
</tr>
<tr>
<td>TrafficSpace</td>
<td>«FeatureType» TrafficSpace</td>
</tr>
<tr>
<td>Waterway</td>
<td>«TopLevelFeatureType» Waterway</td>
</tr>
<tr>
<td>ADEOfAbstractTransportationSpace</td>
<td>«DataType» ADEOfAbstractTransportationSpace</td>
</tr>
<tr>
<td>ADEOf AuxiliaryTrafficArea</td>
<td>«DataType» ADEOfAuxiliaryTrafficArea</td>
</tr>
<tr>
<td>ADEOf AuxiliaryTrafficSpace</td>
<td>«DataType» ADEOfAuxiliaryTrafficSpace</td>
</tr>
<tr>
<td>ADEOf ClearanceSpace</td>
<td>«DataType» ADEOfClearanceSpace</td>
</tr>
<tr>
<td>ADEOf Hole</td>
<td>«DataType» ADEOfHole</td>
</tr>
<tr>
<td>ADEOf Hole Surface</td>
<td>«DataType» ADEOfHoleSurface</td>
</tr>
<tr>
<td>ADEOf Intersection</td>
<td>«DataType» ADEOfIntersection</td>
</tr>
<tr>
<td>ADEOf Marking</td>
<td>«DataType» ADEOfMarking</td>
</tr>
<tr>
<td>ADEOf Railway</td>
<td>«DataType» ADEOfRailway</td>
</tr>
<tr>
<td>ADEOf Road</td>
<td>«DataType» ADEOfRoad</td>
</tr>
<tr>
<td>ADEOf Section</td>
<td>«DataType» ADEOfSection</td>
</tr>
<tr>
<td>ADEOf Square</td>
<td>«DataType» ADEOfSquare</td>
</tr>
<tr>
<td>ADEOf Track</td>
<td>«DataType» ADEOfTrack</td>
</tr>
<tr>
<td>ADEOf Traffic Area</td>
<td>«DataType» ADEOfTrafficArea</td>
</tr>
<tr>
<td>ADEOf Traffic Space</td>
<td>«DataType» ADEOfTrafficSpace</td>
</tr>
<tr>
<td>ADEOf Waterway</td>
<td>«DataType» ADEOfWaterway</td>
</tr>
</tbody>
</table>

| Requirement 34 | /req/transportation/reference |
The following properties only allow to reference other features. The use of these properties is subject to the following restrictions:

A  If the predecessor property (type: gml:ReferenceType) of the TrafficSpace element is not null, it SHALL contain an XLink reference to a TrafficSpace element.

B  If the successor property (type: gml:ReferenceType) of the TrafficSpace element is not null, it SHALL contain an XLink reference to a TrafficSpace element.

Surface boundaries are constrained by the following requirement:

<table>
<thead>
<tr>
<th>Requirement 35</th>
<th>/req/transportation/boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table 24</strong> lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Transportation module. If a space element is bounded by thematic surface boundaries using the property <code>core:boundary</code> (type: <code>core:AbstractSpaceBoundaryPropertyType</code>), the property SHALL contain exactly one surface element from <strong>Table 24</strong> that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 24** lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Transportation module:

**Table 24. Transportation space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tran:AbstractTransportationSpace</code></td>
<td>• <code>tran:Marking</code>&lt;br&gt;• <code>core:ClosureSurface</code>&lt;br&gt;• <code>gen:GenericThematicSurface</code>&lt;br&gt;• possible surface elements from ADEs</td>
</tr>
<tr>
<td><code>tran:AuxiliaryTrafficSpace</code></td>
<td>• <code>tran:AuxiliaryTrafficArea</code>&lt;br&gt;• <code>core:ClosureSurface</code>&lt;br&gt;• <code>gen:GenericThematicSurface</code>&lt;br&gt;• possible surface elements from ADEs</td>
</tr>
<tr>
<td><code>tran:ClearanceSpace</code></td>
<td>• <code>core:ClosureSurface</code>&lt;br&gt;• <code>gen:GenericThematicSurface</code>&lt;br&gt;• possible surface elements from ADEs</td>
</tr>
</tbody>
</table>
| tran:Hole      | • tran:HoleSurface  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
| tran:Intersection | • tran:Marking  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
| tran:Railway   | • tran:Marking  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
| tran:Road      | • tran:Marking  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
| tran:Section   | • tran:Marking  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
| tran:Square    | • tran:Marking  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
| tran:Track     | • tran:Marking  
|               | • core:ClosureSurface  
|               | • gen:GenericThematicSurface  
|               | • possible surface elements from ADEs  |
### 6.15. Tunnel

The Tunnel module supports representation of thematic and spatial aspects of tunnels, tunnel parts, tunnel installations, and interior tunnel structures.

The concepts of the Tunnel module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Tunnel module is available under [https://docs.ogc.org/is/20-010/20-010.html#tunnel-uml](https://docs.ogc.org/is/20-010/20-010.html#tunnel-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc65](https://docs.ogc.org/is/20-010/20-010.html#toc65).

<table>
<thead>
<tr>
<th>Requirements Class</th>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-tunnel">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-tunnel</a></td>
<td>Dependency</td>
<td>tunnel.xsd</td>
</tr>
<tr>
<td></td>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel</a></td>
</tr>
<tr>
<td></td>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

- **Tunnel**: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel)

The applicable GML schema is `tunnel.xsd`.

### 6.15.1. Dependencies

The Tunnel Requirements Class is dependent on the following Requirements Class from this standard:

- The **Core** Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.
6.15.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference.

6.15.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 36</th>
<th>/req/tunnel/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in tunnel.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Tunnel XML elements and their corresponding CityGML UML classes are listed in Table 25. In addition, the CityGML Tunnel XML elements are documented in the GML schema tunnel.xsd and in this XML Schema documentation.

**Table 25. Tunnel XML Elements**

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractTunnel</td>
<td>«FeatureType» AbstractTunnel</td>
</tr>
<tr>
<td>HollowSpace</td>
<td>«FeatureType» HollowSpace</td>
</tr>
<tr>
<td>Tunnel</td>
<td>«TopLevelFeatureType» Tunnel</td>
</tr>
<tr>
<td>TunnelConstructiveElement</td>
<td>«FeatureType» TunnelConstructiveElement</td>
</tr>
<tr>
<td>TunnelFurniture</td>
<td>«FeatureType» TunnelFurniture</td>
</tr>
<tr>
<td>TunnelInstallation</td>
<td>«FeatureType» TunnelInstallation</td>
</tr>
<tr>
<td>TunnelPart</td>
<td>«FeatureType» TunnelPart</td>
</tr>
<tr>
<td>ADEOfAbstractTunnel</td>
<td>«DataType» ADEOfAbstractTunnel</td>
</tr>
<tr>
<td>ADEOfHollowSpace</td>
<td>«DataType» ADEOfHollowSpace</td>
</tr>
<tr>
<td>ADEOfTunnel</td>
<td>«DataType» ADEOfTunnel</td>
</tr>
<tr>
<td>ADEOfTunnelConstructiveElement</td>
<td>«DataType» ADEOfTunnelConstructiveElement</td>
</tr>
<tr>
<td>ADEOfTunnelFurniture</td>
<td>«DataType» ADEOfTunnelFurniture</td>
</tr>
<tr>
<td>ADEOfTunnelInstallation</td>
<td>«DataType» ADEOfTunnelInstallation</td>
</tr>
<tr>
<td>ADEOfTunnelPart</td>
<td>«DataType» ADEOfTunnelPart</td>
</tr>
</tbody>
</table>

Surface boundaries are constrained by the following requirement:

<table>
<thead>
<tr>
<th>Requirement 37</th>
<th>/req/tunnel/boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 26 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Tunnel module. If a space element is bounded by thematic surface boundaries using the property `core:boundary` (type: `core:AbstractSpaceBoundaryPropertyType`), the property SHALL contain exactly one surface element from Table 26 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

Table 26 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Tunnel module:

**Table 26. Tunnel space elements and their allowed thematic surface boundaries**

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
</table>
| `tun:AbstractTunnel` | • `con:AbstractConstructionSurface` and the substitutable elements:  
                        • `con:GroundSurface`,  
                        • `con:RoofSurface`,  
                        • `con:CeilingSurface`,  
                        • `con:OuterCeilingSurface`,  
                        • `con:FloorSurface`,  
                        • `con:OuterFloorSurface`,  
                        • `con:WallSurface`,  
                        • `con:InteriorWallSurface`  
                        • `core:ClosureSurface`  
                        • `gen:GenericThematicSurface`  
                        • possible surface elements from ADEs |
| tun:Tunnel                | • con:AbstractConstructionSurface and the substitutable elements:  
|                          |   con:GroundSurface,  
|                          |   con:RoofSurface,  
|                          |   con:CeilingSurface,  
|                          |   con:OuterCeilingSurface,  
|                          |   con:FloorSurface,  
|                          |   con:OuterFloorSurface,  
|                          |   con:WallSurface,  
|                          |   con:InteriorWallSurface  
|                          | • core:ClosureSurface  
|                          | • gen:GenericThematicSurface  
|                          | • possible surface elements from ADEs  
| tun:TunnelConstractiveElement | • con:AbstractConstructionSurface and the substitutable elements:  
|                              |   con:GroundSurface,  
|                              |   con:RoofSurface,  
|                              |   con:CeilingSurface,  
|                              |   con:OuterCeilingSurface,  
|                              |   con:FloorSurface,  
|                              |   con:OuterFloorSurface,  
|                              |   con:WallSurface,  
|                              |   con:InteriorWallSurface  
|                              | • core:ClosureSurface  
|                              | • gen:GenericThematicSurface  
|                              | • possible surface elements from ADEs  
| tun:TunnelFurniture       | No boundaries allowed  
| tun:TunnelInstallation   | • con:AbstractConstructionSurface and the substitutable elements:  
|                          |   con:GroundSurface,  
|                          |   con:RoofSurface,  
|                          |   con:CeilingSurface,  
|                          |   con:OuterCeilingSurface,  
|                          |   con:FloorSurface,  
|                          |   con:OuterFloorSurface,  
|                          |   con:WallSurface,  
|                          |   con:InteriorWallSurface  
|                          | • core:ClosureSurface  
|                          | • gen:GenericThematicSurface  
|                          | • possible surface elements from ADEs  

6.16. Vegetation

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

The concepts of the Vegetation module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Vegetation module is available under https://docs.ogc.org/is/20-010/20-010.html#vegetation-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc59.

### Requirements Class

<table>
<thead>
<tr>
<th>Requirement Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-vegetation">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-vegetation</a></td>
<td>Implementation</td>
</tr>
<tr>
<td>Dependency</td>
<td>vegetation.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-vegetation">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-vegetation</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

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

The applicable GML schema is vegetation.xsd.

### 6.16.1. Dependencies

The Vegetation Requirements Class is dependent on the following Requirements Class from this standard:
• The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

### 6.16.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference.

### 6.16.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 38</th>
<th>/req/vegetation/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in vegetation.xsd.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Vegetation XML elements and their corresponding CityGML UML classes are listed in Table 27. In addition, the CityGML Vegetation XML elements are documented in the GML schema vegetation.xsd and in this XML Schema documentation.

#### Table 27. Vegetation XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractVegetationObject</td>
<td>«FeatureType» AbstractVegetationObject</td>
</tr>
<tr>
<td>PlantCover</td>
<td>«TopLevelFeatureType» PlantCover</td>
</tr>
<tr>
<td>SolitaryVegetationObject</td>
<td>«TopLevelFeatureType» SolitaryVegetationObject</td>
</tr>
<tr>
<td>ADEOfAbstractVegetationObject</td>
<td>«DataType» ADEOfAbstractVegetationObject</td>
</tr>
<tr>
<td>ADEOfPlantCover</td>
<td>«DataType» ADEOfPlantCover</td>
</tr>
<tr>
<td>ADEOfSolitaryVegetationObject</td>
<td>«DataType» ADEOfSolitaryVegetationObject</td>
</tr>
</tbody>
</table>

Surface boundaries are constrained by the following requirement:

<table>
<thead>
<tr>
<th>Requirement 39</th>
<th>/req/vegetation/boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 28 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Vegetation module. If a space element is bounded by thematic surface boundaries using the property core:boundary (type: core:AbstractSpaceBoundaryPropertyType), the property SHALL contain exactly one surface element from Table 28 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.</td>
<td></td>
</tr>
</tbody>
</table>

Table 28 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Vegetation module.
Table 28. Vegetation space elements and their allowed thematic surface boundaries

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>veg:AbstractVegetationObject</td>
<td>No boundaries allowed</td>
</tr>
<tr>
<td>veg:PlantCover</td>
<td>No boundaries allowed</td>
</tr>
<tr>
<td>veg:SolitaryVegetationObject</td>
<td>No boundaries allowed</td>
</tr>
</tbody>
</table>

6.17. Versioning

The Versioning module supports representation of multiple versions of CityGML features within a single CityGML model. In addition, the version transitions and transactions that lead to the different versions can be represented.

The concepts of the Versioning module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the Versioning module is available under https://docs.ogc.org/is/20-010/20-010.html#versioning-uml, the Data Dictionary is available under https://docs.ogc.org/is/20-010/20-010.html#toc60.

<table>
<thead>
<tr>
<th>Requirements Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-versioning">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-versioning</a></td>
<td></td>
</tr>
<tr>
<td>Target type</td>
<td>Implementation</td>
</tr>
<tr>
<td>Dependency</td>
<td>versioning.xsd</td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-versioning">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-versioning</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

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

The applicable GML schema is versioning.xsd.

6.17.1. Dependencies

The Versioning Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.
6.17.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference. However, restrictions are defined for the following associations, since they represent alternative aggregations (see Requirement /req/global/alternativeaggregations on referencing features from alternative aggregations):

   ◦ The association `oldFeature` which connects `Transaction` with `AbstractFeatureWithLifespan`; here, a transaction must reference the version of the city object prior to the transaction using XLink.
   
   ◦ The association `newFeature` which connects `Transaction` with `AbstractFeatureWithLifespan`; here, a transaction must reference the version of the city object subsequent to the transaction using XLink.
   
   ◦ The association `from which` connects `VersionTransition` with `Version`; here, a version transition must reference the predecessor version of the `VersionTransition` using XLink.
   
   ◦ The association `to which` connects `VersionTransition` with `Version`; here, a version transition must reference the successor version of the `VersionTransition` using XLink.
   
   ◦ The association `versionMember` which connects `Version` with `AbstractFeatureWithLifespan`; here, a version must reference the features that are part of this version using XLink.

6.17.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 40</th>
<th>/req/versioning/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in <code>versioning.xsd</code>.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML Versioning XML elements and their corresponding CityGML UML classes are listed in Table 29. In addition, the CityGML Versioning XML elements are documented in the GML schema `versioning.xsd` and in this XML Schema documentation.

Table 29. Versioning XML Elements

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>«FeatureType» Version</td>
</tr>
<tr>
<td>VersionTransition</td>
<td>«FeatureType» VersionTransition</td>
</tr>
<tr>
<td>ADEOfVersion</td>
<td>«DataType» ADEOfVersion</td>
</tr>
<tr>
<td>ADEOfVersionTransition</td>
<td>«DataType» ADEOfVersionTransition</td>
</tr>
<tr>
<td>Transaction</td>
<td>«DataType» Transaction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirement 41</th>
<th>/req/versioning/reference</th>
</tr>
</thead>
</table>
The following properties only allow to reference other features. The use of these properties is subject to the following restrictions:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>If the <code>from</code> property (type: <code>gml:ReferenceType</code>) of the <code>VersionTransition</code> element is not null, it SHALL contain an XLink reference to a <code>Version</code> element.</td>
</tr>
<tr>
<td>B</td>
<td>If the <code>to</code> property (type: <code>gml:ReferenceType</code>) of the <code>VersionTransition</code> element is not null, it SHALL contain an XLink reference to a <code>Version</code> element.</td>
</tr>
<tr>
<td>C</td>
<td>If the <code>oldFeature</code> property (type: <code>gml:ReferenceType</code>) of the <code>Transaction</code> element is not null, it SHALL contain an XLink reference to a <code>core:AbstractFeatureWithLifespan</code> element.</td>
</tr>
<tr>
<td>D</td>
<td>If the <code>newFeature</code> property (type: <code>gml:ReferenceType</code>) of the <code>Transaction</code> element is not null, it SHALL contain an XLink reference to a <code>core:AbstractFeatureWithLifespan</code> element.</td>
</tr>
<tr>
<td>E</td>
<td>If the <code>versionMember</code> property (type: <code>gml:AbstractFeatureMemberType</code>) of the <code>Version</code> element is not null, it SHALL contain an XLink reference to a <code>core:AbstractFeatureWithLifespan</code> element.</td>
</tr>
</tbody>
</table>

### 6.18. WaterBody

The WaterBody module supports representation of the thematic aspects and 3D geometry of rivers, canals, lakes, and basins. It does, however, not inherit any hydrological or other dynamic aspects of fluid flow.

The concepts of the WaterBody module are defined in the OGC CityGML 3.0 Conceptual Model standard. The UML diagram for the WaterBody module is available under [https://docs.ogc.org/is/20-010/20-010.html#waterbody-uml](https://docs.ogc.org/is/20-010/20-010.html#waterbody-uml), the Data Dictionary is available under [https://docs.ogc.org/is/20-010/20-010.html#toc61](https://docs.ogc.org/is/20-010/20-010.html#toc61).

**Requirements Class**

```
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-waterbody
```

<table>
<thead>
<tr>
<th>Target type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency</td>
<td><code>waterBody.xsd</code></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody">http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody</a></td>
</tr>
<tr>
<td>Dependency</td>
<td><a href="http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core">http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core</a></td>
</tr>
</tbody>
</table>

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

- **WaterBody**: [http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody](http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody)

The applicable GML schema is `waterBody.xsd`. 
6.18.1. Dependencies

The WaterBody Requirements Class is dependent on the following Requirements Class from this standard:

- The Core Requirements Class which is the only mandatory Requirements Class. This class contains foundational elements upon which a CityGML dataset is built. The Core Requirements Class also defines XML elements and types reused by other Requirements Classes.

6.18.2. Implementation Decisions

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

1. All associations allow by default that the referenced features can be provided inline or by reference.

6.18.3. Requirements

<table>
<thead>
<tr>
<th>Requirement 42</th>
<th>/req/waterbody/elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityGML XML elements implemented by a conforming instance document shall conform to the GML XML schema in <code>waterBody.xsd</code>.</td>
<td></td>
</tr>
</tbody>
</table>

The CityGML WaterBody XML elements and their corresponding CityGML UML classes are listed in **Table 30**. In addition, the CityGML WaterBody XML elements are documented in the GML schema `waterBody.xsd` and in this XML Schema documentation.

**Table 30. WaterBody XML Elements**

<table>
<thead>
<tr>
<th>XML Schema Element (XML Schema documentation)</th>
<th>Conceptual Model (UML model; Data Dictionary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractWaterBoundarySurface</td>
<td>«FeatureType» AbstractWaterBoundarySurface</td>
</tr>
<tr>
<td>WaterBody</td>
<td>«TopLevelFeatureType» WaterBody</td>
</tr>
<tr>
<td>WaterGroundSurface</td>
<td>«FeatureType» WaterGroundSurface</td>
</tr>
<tr>
<td>WaterSurface</td>
<td>«FeatureType» WaterSurface</td>
</tr>
<tr>
<td>ADEOfAbstractWaterBoundarySurface</td>
<td>«DataType» ADEOfAbstractWaterBoundarySurface</td>
</tr>
<tr>
<td>ADEOfWaterBody</td>
<td>«DataType» ADEOfWaterBody</td>
</tr>
<tr>
<td>ADEOfWaterGroundSurface</td>
<td>«DataType» ADEOfWaterGroundSurface</td>
</tr>
<tr>
<td>ADEOfWaterSurface</td>
<td>«DataType» ADEOfWaterSurface</td>
</tr>
</tbody>
</table>

Surface boundaries are constrained by the following requirement:

<table>
<thead>
<tr>
<th>Requirement 43</th>
<th>/req/waterbody/boundaries</th>
</tr>
</thead>
</table>
Table 31 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the WaterBody module. If a space element is bounded by thematic surface boundaries using the property core:boundary (type: core:AbstractSpaceBoundaryPropertyType), the property SHALL contain exactly one surface element from Table 31 that is supported for the specific space element. If no surface element is supported, the space element SHALL NOT be bounded by thematic surface boundaries.

<table>
<thead>
<tr>
<th>Space element</th>
<th>Allowed space boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• core:ClosureSurface</td>
</tr>
<tr>
<td></td>
<td>• gen:GenericThematicSurface</td>
</tr>
<tr>
<td></td>
<td>• possible surface elements from ADEs</td>
</tr>
</tbody>
</table>
Chapter 7. Media Types

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

The registered MIME Media Type for GML documents is application/gml+xml.
Annex A: Conformance Class Abstract Test Suite (Normative)

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

A.1. Conformance Class A

A.1.1. Requirement 1

<table>
<thead>
<tr>
<th>Test id:</th>
<th>/conf/conf-class-a/req-name-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement:</td>
<td>/req/req-class-a/req-name-1</td>
</tr>
<tr>
<td>Test purpose:</td>
<td>Verify that...</td>
</tr>
<tr>
<td>Test method:</td>
<td>Inspect...</td>
</tr>
</tbody>
</table>

A.1.2. Requirement 2
Annex B: Examples (Informative)

Various CityGML 3.0 test data sets are available on GitHub for each CityGML module, including those test data sets presented in this chapter: https://github.com/opengeospatial/CityGML-3.0-Encodings/tree/master/CityGML/Examples.

The examples and information provided in this chapter focus on the new CityGML 3.0 modules Dynamizer, PointCloud, and Versioning. In addition, some examples for the description of addresses using xAL 3 are provided (this is a change from CityGML 2.0, where xAL 2 was used). Since the ADE mechanism has been improved in CityGML 3.0, examples for the use of the revised ADE mechanism are provided as well.

B.1. Dynamizer Examples

This chapter provides examples on the use of the Dynamizer concept to integrate dynamic data coming from sensors or being stored in files with city models. A Dynamizer is a feature that can be included in any CityGML thematic feature (e.g. Building, WallSurface, Section of a road) and that provides a pointer to a specific property of the respective thematic feature, whose static value shall be overridden by timeseries data. See Chapter 6.6 of the CityGML Conceptual Model Standard and Chaturvedi & Kolbe 2019 for more information. The examples provided here have been taken from Chaturvedi 2021.

A Dynamizer can provide the timeseries data in different ways:

- As a link to a specific datastream provided by a sensor data service like OGC SensorThings API, OGC Sensor Observation Service, or other sensor data platforms (incl. MQTT). For this purpose the Dynamizer makes use of a SensorConnection object;
- As a reference to an external file with tabulated data (e.g. CSV, MS Excel, Google Spreadsheet) for which the Dynamizer uses a TabulatedFileTimeseries object;
- As a reference to an external file formatted according to OGC TimeseriesML or Observations & Measurements standards using a StandardFileTimeseries object;
- Inline within the Dynamizer feature using a GenericTimeseries object;
- By defining CompositeTimeseries objects which consist of a combination of various GenericTimeseries objects, TabulatedFileTimeseries objects, and/or StandardFileTimeseries objects.

For each of these possibilities, examples are provided in the following.

B.1.1. Examples for integrating city objects with real-time sensors

The following examples refer to a Building feature shown in Listing 4. The building has a roof with the ID roof1 and a room with the ID room1. The building defines two generic attributes that provide a) a static value for the temperature on the roof (generic attribute outside_temperature) and b) a static value for the temperature in the room (generic attribute temperature). In addition, the building might also have other static attributes such as building function, roof type, and address which are not shown here.
The room has an indoor sensor installed which measures the air temperature and humidity in the room. The observations from this sensor can be used to override the static value of the generic attribute `temperature` with dynamic values coming from the sensor.

In addition, a personal weather station is installed on the roof which measures weather-related properties such as outside temperature, relative humidity, and wind speed.

Listing 4. A Building with a RoofSurface containing a generic attribute 'outside_temperature' and a BuildingRoom containing a generic attribute 'temperature'.

```
<core:cityObjectMember>
  <bldg:Building gml:id="building1">
    <core:boundary>
      <con:RoofSurface gml:id="roof1">
        <core:genericAttribute>
          <gen:DoubleAttribute>
            <gen:name>outside_temperature</gen:name>
            <gen:value>23.3</gen:value>
          </gen:DoubleAttribute>
        </core:genericAttribute>
      </con:RoofSurface>
    </core:boundary>
    <bldg:buildingRoom>
      <bldg:BuildingRoom gml:id="room1">
        <core:genericAttribute>
          <gen:DoubleAttribute>
            <gen:name>temperature</gen:name>
            <gen:value>20.2</gen:value>
          </gen:DoubleAttribute>
        </core:genericAttribute>
      </bldg:BuildingRoom>
    </bldg:buildingRoom>
  </bldg:Building>
</core:cityObjectMember>
```

In general, real-time sensor data streams are available from numerous platforms. In the following, examples are provided for the use of the ThingSpeak platform, the OGC SensorThings API, and the MQTT protocol to retrieve the room temperature from the indoor sensor, as well as the WeatherUnderground Network and the OGC Sensor Observation Service to retrieve the outside temperature data from the weather station.

B.1.1.1. Example 1: ThingSpeak platform

The following is an example query to request sensor data from the ThingSpeak platform (the request is a single line without linebreaks and spaces). The request retrieves all observations between '2019-10-08T00:00:00Z' and '2019-10-09T00:00:00Z' for the field ID 1. ThingSpeak manages each registered sensor using a unique channel ID (here: 64242). A query to the channel ID delivers details on the sensor device and on the available datastreams. In this example, the datastreams available are temperature and humidity registered with the field IDs 1 and 2, respectively:
Listing 5 shows, how the Dynamizer makes use of this query to link the generic attribute \textit{temperature} to the temperature observation of the ThingSpeak datastream with field ID 1.

\textbf{Listing 5. Dynamizer defining a sensor connection to the ThingSpeak platform.}

```xml
<bldg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer1">
      <dyn:attributeRef>
        //bldg:BuildingRoom[@gml:id='room1']/core:genericAttribute/
gen:DoubleAttribute[gen:name='temperature']/gen:value
      </dyn:attributeRef>
      <dyn:startTime>2019-10-08T00:00:00Z</dyn:startTime>
      <dyn:endTime>2019-10-09T00:00:00Z</dyn:endTime>
      <dyn:sensorConnection>
        <dyn:SensorConnection>
          <dyn:connectionType>thingspeak_1.0</dyn:connectionType>
          <dyn:observationProperty>Temperature</dyn:observationProperty>
          <dyn:uom>Celsius</dyn:uom>
          <dyn:sensorID>64242</dyn:sensorID>
          <dyn:sensorName>DHT22</dyn:sensorName>
          <dyn:datastreamID>1</dyn:datastreamID>
          <dyn:authType>none</dyn:authType>
          <dyn:linkToObservation>
            <!--Single line without linebreaks and spaces-->
            %baseURL%/channels/%sensorID%/fields/%datastreamID%.json?start=%startTime%&end=%endTime%
          </dyn:linkToObservation>
          <dyn:sensorLocation xlink:href="#room1"/>
        </dyn:SensorConnection>
      </dyn:sensorConnection>
    </dyn:Dynamizer>
  </core:dynamizer>
</bldg:Building>
```

The Dynamizer specifies the following parameters:

- \textit{attributeRef} defines an XPath expression which refers to the generic attribute \textit{temperature} of the BuildingRoom feature with the ID \textit{room1} that is to be overridden by sensor data from the specific ThingSpeak channel.
- \textit{startTime} and \textit{endTime} represent the entire temporal extent for the sensor data to be retrieved.
The actual connection parameters to the ThingSpeak platform are specified by the Dynamizer using a SensorConnection object:

- **connectionType** indicates the type of the connection, which is `thingspeak_1.0` in this example.
- **observationProperty** indicates the observed property for which sensor data is retrieved.
- **uom** specifies the unit of measurement of the observed property.
- **datastreamID** contains the field ID of the datastream to be retrieved.
- **baseURL** specifies the base URL for all channels available on the ThingSpeak platform. The value is always [https://thingspeak.com](https://thingspeak.com).
- **sensorID** provides the ID of the sensor.
- **sensorName** provides the name of the sensor, in this example it is a DHT22 sensor.
- **authType** specifies the authentication required to connect to the sensor. In this example, the parameter value is `none`, since the sensor is publicly available and requires no authentication.
- **linkToObservation** allows for encoding the URL to establish a direct link to the observations of the specific datastream. The URL is a combination of static content and placeholders. In this example the following placeholders are defined: `%baseURL%`, `%sensorID%`, `%datastreamID%`, `%startTime%`, and `%endTime%`. The complete URL is generated by substituting the placeholders with the values from the corresponding parameters defined by the Dynamizer and the SensorConnection. In this example the resulting URL is: [https://thingspeak.com/channels/64242/fields/1?start=2019-10-08T00:00:00Z&end=2019-10-09T00:00:00Z](https://thingspeak.com/channels/64242/fields/1?start=2019-10-08T00:00:00Z&end=2019-10-09T00:00:00Z).
- **sensorLocation** indicates where the sensor is located, which is the BuildingRoom with the ID `room1` in this example.

### B.1.1.2. Example 2: OGC SensorThings API

In this example, the same room temperature from sensor DHT22 is to be retrieved, but now the sensor observations are stored in a FROST Server and, thus, need to be queried using the OGC SensorThings API. The following is an example query to request sensor data from the FROST Server (the request is a single line without linebreaks and spaces). The request retrieves all observations between '2019-04-12T10:00:00.000Z' and '2019-04-12T10:30:00.000Z' for the Datastream ID 1 which is the temperature. According to the SensorThings API, observations are always aligned with datastreams:

```
http://127.0.0.1:8080/FROST-Server/v1.0/Datastreams(1)/Observations?$filter=during(phenomenonTime,2019-04-12T10:00:00.000Z/2019-04-12T10:30:00.000Z)
```

Listing 6 shows, how the Dynamizer needs to be defined to retrieve observations from the FROST Server and link them to the generic attribute `temperature`. 
The parameters defined by the Dynamizer itself are identical to those shown in the example for the ThingSpeak platform.

However, the following parameters provided within the SensorConenction object differ, since the Dynamizer defines now a connection to the FROST Server:

- **connectionType** is set to `ogc_sta_1.0`, which represents the OGC SensorThings API version 1.0.
- **baseURL** is set to the URL for accessing a specific FROST Server. The base URL may differ for FROST Servers running on different machines or cloud environments.
- **linkToObservation** makes use again of the combination of static content and placeholders to define the URL for retrieving the observations. The following placeholders are defined:
%baseURL%, %datastreamID%, %startTime%, and %endTime%. Attribute substitution results in the following URL: http://127.0.0.1:8080/FROSTServer/v1.0/Datastreams(1)/Observations?$filter=during(phenomenonTime,2019-04-12T10:00:00.000Z/2019-04-12T10:30:00.000Z).

- linkToSensorDescription allows to retrieve the sensor details corresponding to the specified datastream. Also, here attribute substitution is used to obtain the URL http://127.0.0.1:8080/FROSTServer/v1.0/Datastreams(1)/Sensor.

The parameters observationProperty, uom, datastreamID, authType, and sensorLocation remain unchanged, as they specify the same values as in the example for the ThingSpeak platform. In contrast to connecting to the ThingSpeak platform, the parameter sensorID is not required for SensorThings API connections; the datastream ID is sufficient for this purpose.

### B.1.1.3. Example 3: Subscribing to real-time datatreams using the MQTT protocol

In this example, the room temperature from sensor DHT22 is to be retrieved again, but this time, the observations are available on the Things Network platform. The Things Network supports the MQTT protocol and allows users to subscribe to its data streams. The following is an example request to connect to the Things Network via MQTT. Server represents the URL of the machine where the server is deployed. Topic connects to the individual datastream. The authentication details are provided as username and password. These details are required as separate attributes by an MQTT broker; they cannot be provided within a single URL:

```
Server: eu.thethings.network
Topic: +/devices/tumgis-dragino-shield-with-gps/up/temperature
Username: **********
Password: **********
```

Listing 7 shows, how the Dynamizer is defined to link the generic attribute temperature to the temperature observations using MQTT.
Listing 7. Dynamizer subscribing to a sensor data stream using the MQTT protocol.

```xml
<blgd:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer3">
      <dyn:attributeRef>
        <!--Single line without linebreaks and spaces-->
        //blgd:BuildingRoom[@gml:id='room1']/core:genericAttribute/
        gen:DoubleAttribute[gen:name='temperature']/gen:value
      </dyn:attributeRef>
      <dyn:startTime>2019-01-01T00:00:00.000Z</dyn:startTime>
      <dyn:endTime>2020-01-01T00:00:00.000Z</dyn:endTime>
      <dyn:sensorConnection>
        <dyn:SensorConnection>
          <dyn:connectionType>mqtt_3.1.1</dyn:connectionType>
          <dyn:observationProperty>Temperature</dyn:observationProperty>
          <dyn:uom>Celsius</dyn:uom>
          <dyn:authType>basic</dyn:authType>
          <dyn:mqttServer>eu.thethings.network</dyn:mqttServer>
          <dyn:mqttTopic>
            +/devices/tumgis-dragino-shield-with-gps/up/temperature
          </dyn:mqttTopic>
          <dyn:sensorLocation xlink:href="#room1"/>
        </dyn:SensorConnection>
      </dyn:sensorConnection>
    </dyn:Dynamizer>
  </core:dynamizer>
</blgd:Building>
```

The following parameters and parameter values of the SensorConnection object are specific to the MQTT protocol:

- **connectionType** in this case is `mqtt_3.1.1`. It shows the version of the MQTT protocol as 3.1.1.
- **authType** is indicated as `basic`, which means that a basic authentication (with username and password) is required to establish the connection.
- **mqttServer** provides the name of the MQTT server.
- **mqttTopic** provides the specific datastream that is to be retrieved.

### B.1.1.4. Example 4: Weather Underground API with authentication

In this example, the outside temperature measured by the weather station on the roof of the building is to be retrieved. The weather station is registered on the Weather Underground platform with the identifier 'MyPWS1'. The following is an example query to request sensor data from the Weather Underground API. The Weather Underground API requires a unique API Key for every user allowing to authenticate to a running sensor ID:
Listing 8 shows the Dynamizer definition required to connect the generic attribute *outside_temperature* to the weather station registered on the Weather Underground platform.

Listing 8. Dynamizer defining a sensor connection to the Weather Underground platform.

```xml
<bldg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer4">
      <!--Single line without linebreaks and spaces-->
      //con:RoofSurface[@gml:id='roof1']/core:genericAttribute/
      gen:DoubleAttribute[gen:name='outside_temperature']/gen:value
    </dyn:Dynamizer>
    <dyn:startTime>2019-01-01T10:00:00.000Z</dyn:startTime>
    <dyn:endTime>2019-02-01T00:00:00.000Z</dyn:endTime>
    <dyn:sensorConnection>
      <dyn:SensorConnection>
        <dyn:connectionType>wunderground_1.0</dyn:connectionType>
        <dyn:observationProperty>Temperature</dyn:observationProperty>
        <dyn:uom>Celsius</dyn:uom>
        <dyn:sensorID>MyPWS1</dyn:sensorID>
        <dyn:baseURL>
          http://api.wunderground.com/api
        </dyn:baseURL>
        <dyn:authType>apiKey</dyn:authType>
        <dyn:linkToObservation>%baseUrl%/%authType%/conditions/q/pws:%sensorID%.json
        </dyn:linkToObservation>
        <dyn:sensorLocation xlink:href="#roof1"/>
      </dyn:SensorConnection>
    </dyn:sensorConnection>
  </core:dynamizer>
  <!-- The roof surface and building room from {Listing 4} come here -->
</bldg:Building>
```

The following parameters of the Dynamizer and the SensorConnection object differ from the examples above:

- *attributeRef* defines now an XPath expression to refer to the value of the generic attribute *outside_temperature* of the RoofSurface with the ID *roof1*.
- *connectionType* is set to *wunderground_1.0*.
- *sensorID* is set to *MyPWS1*.
- *baseURL* is *http://api.wunderground.com/api*; the value is the same for all requests to the WeatherUnderground API.
• **authType** has the value *apiKey*, because the API requires a unique API Key to connect to the server.

• **linkToObservation** uses now the following placeholders: %baseUrl%, %authType%, and %sensorID%.

• **sensorLocation** indicates that the sensor is located on the roof of the building.

### B.1.1.5. Example 5: OGC Sensor Observation Service

In this example, the same outside temperature from the weather station is to be retrieved, but now the sensor observations are stored in an OGC Sensor Observation Service (SOS). The observations can then be retrieved using the standardised SOS operation *GetObservation*. The following is an example *GetObservation* request (the request is a single line without linebreaks and spaces). The request retrieves all temperature observations between '2019-10-12T10:00:00.000Z' and '2019-10-09T00:00:00Z':

```plaintext
http://127.0.0.1:8080/weather-sensors-sos-webapp/service?SERVICE=SOS&VERSION=2.0.0&REQUEST=GetObservation&PROCEDURE=MyPWS1&OBSERVEDPROPERTY=Temp&TEMPORALFILTER=om:phenomenonTime,2019-10-08T00:00:00Z/2019-10-09T00:00:00Z
```

Listing 9 shows, how the Dynamizer needs to be defined to retrieve observations from the OGC SOS and link them to the generic attribute outside_temperature.
Listing 9. Dynamizer defining a sensor connection to an OGC Sensor Observation Service.

```xml
<bdg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer5">
      <dyn:attributeRef>
        <!-- Single line without linebreaks and spaces -->
        //con:RoofSurface[@gml:id='roof1']/core:genericAttribute/
        gen:DoubleAttribute[gen:name='outside_temperature']/gen:value
      </dyn:attributeRef>
      <dyn:startTime>2019-01-01T00:00:00Z</dyn:startTime>
      <dyn:endTime>2020-01-01T00:00:00Z</dyn:endTime>
      <dyn:sensorConnection>
        <dyn:SensorConnection>
          <dyn:connectionType>ogc_sos_2.0</dyn:connectionType>
          <dyn:observationProperty>Temperature</dyn:observationProperty>
          <dyn:uom>Celsius</dyn:uom>
          <dyn:sensorID>MyPWS1</dyn:sensorID>
          <dyn:baseURL>
            http://127.0.0.1:8080/weather-sensors-sos-webapp/service
          </dyn:baseURL>
          <dyn:authType>none</dyn:authType>
          <dyn:linkToObservation>
            <!-- Single line without linebreaks and spaces -->
            %baseURL%/request=GetObservation&service=SOS&version=2.0.0
            &procedure=%sensorID%&observedProperty=%observationProperty%
            &temporalFilter=om:phenomenonTime,%startTime%/%endTime%
          </dyn:linkToObservation>
          <dyn:linkToSensorDescription>
            <!-- Single line without linebreaks and spaces -->
            %baseURL%/REQUEST=DescribeSensor&SERVICE=SOS&VERSION=2.0.0
            &PROCEDURE=%sensorID%&procedureDescriptionFormat=http://www.opengis.net/sensorML/1.0.1
          </dyn:linkToSensorDescription>
          <dyn:sensorLocation xlink:href="#roof1"/>
        </dyn:SensorConnection>
      </dyn:sensorConnection>
    </dyn:Dynamizer>
  </core:dynamizer>
</bdg:Building>
```

The following parameters of the Dynamizer and the SensorConnection object differ from the example above:

- **connectionType** is set to **ogc_sos_2.0**.
- **baseURL** is set to the URL for accessing a specific SOS instance. The base URL may differ for SOS instances running on different machines.
- **linkToObservation** uses now the following placeholders: %baseUrl%, %sensorID%,
%observationProperty%, %startTime%, and %endTime%.

- linkToSensorDescription allows to retrieve the sensor details. Here, attribute substitution is used to obtain the URL http://127.0.0.1:8080/weather-sensors-sos-webapp/service?REQUEST= DescribeSensor&SERVICE=SOS&VERSION=2.0.0&PROCEDURE=ILONDON577&procedureDescriptionFormat=http://www.opengis.net/sensorML/1.0.1.

B.1.2. Examples for representing timeseries data inline with city objects or in external files

The following examples show how timeseries data resulting from a solar potential simulation can be stored in tabulated files such as external CSV files or be represented inline with a city object.

The examples refer to a building wall surface with ID wall1 that has a generic attribute globalRadMonth, see Listing 10. The attribute represents the global irradiation value on the wall surface for different months which is calculated by a solar potential simulation. This means that the attribute is dynamic in nature and changes its value every month based on the simulation results.

Listing 10. A Building with a WallSurface containing a generic attribute 'globalRadMonth' to record monthly solar irradiation values.

```
<con:WallSurface gml:id="wall1">
  <core:genericAttribute>
    <gen:DoubleAttribute>
      <gen:name>globalRadMonth</gen:name>
      <gen:value>4293.446</gen:value>
    </gen:DoubleAttribute>
  </core:genericAttribute>
</con:WallSurface>
```

B.1.2.1. Example 1: Representation of timeseries data in tabulated files

In this example, the Dynamizer makes use of the TabulatedFileTimeseries object to link the dynamic data in tabulated files with the attribute of the city object.

Listing 11 shows the CSV file named results.csv that stores the monthly simulation results for the building wall surface. The CSV file has a header line defining the column names. The first column Surface_ID contains the gml:id of the building surfaces. The second column Time contains the timestamp for each irradiation value in the form YYYY-MM. For example, 2015-01 represents the irradiation value computed for January 2015. The next column Uom represents the unit of measurement, which is kilowatt-hour (kWh) in this example. The last three columns Diffuse, Direct, and Global represent the computed diffuse, direct, and global irradiation values for the individual building surfaces. In the given CSV file, the field separator character is the comma (,) and the decimal symbol is the dot (.)
Listing 11. CSV file ‘results.csv’ that stores the monthly solar irradiation values for different building surfaces.

<table>
<thead>
<tr>
<th>Surface_ID</th>
<th>Time</th>
<th>Uom</th>
<th>Diffuse</th>
<th>Direct</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>wall1</td>
<td>2015-01</td>
<td>kWh</td>
<td>1454.653</td>
<td>3315.214</td>
<td>4293.446</td>
</tr>
<tr>
<td>wall1</td>
<td>2015-02</td>
<td>kWh</td>
<td>1866.883</td>
<td>4002.232</td>
<td>5563.502</td>
</tr>
<tr>
<td>wall1</td>
<td>2015-12</td>
<td>kWh</td>
<td>1341.543</td>
<td>3001.412</td>
<td>4010.239</td>
</tr>
<tr>
<td>wall2</td>
<td>2015-01</td>
<td>kWh</td>
<td>1313.344</td>
<td>3112.122</td>
<td>4109.742</td>
</tr>
</tbody>
</table>

Listing 12 shows how the Dynamizer makes use of a TabulatedFileTimeseries object to relate the CSV file results.csv and the individual columns within the file to the generic attribute globalRadMonth.
Listed 12. Dynamizer using a TabulatedFileTimeseries object for referring to timeseries stored in an external CSV file using the column ids.

```xml
<dyn:Dynamizer gml:id="global_irradiation_Dynamizer">
  <dyn:attributeRef>
    <!--Single line without linebreaks and spaces-->
    //con:RoofSurface[@gml:id='wall1']/core:genericAttribute
    /gen:DoubleAttribute[gen:name='globalRadMonth']/gen:value
  </dyn:attributeRef>
  <dyn:startTime>2015-01-01T00:00:00Z</dyn:startTime>
  <dyn:endTime>2016-01-01T00:00:00Z</dyn:endTime>
  <dyn:dynamicData>
    <dyn:TabulatedFileTimeseries>
      <dyn:firstTimestamp>2015-01-01T00:00:00Z</dyn:firstTimestamp>
      <dyn:lastTimestamp>2016-01-01T00:00:00Z</dyn:lastTimestamp>
      <dyn:observationProperty>GlobalIrradiationPerMonth</dyn:observationProperty>
      <dyn:uom>kWh</dyn:uom>
      <dyn:fileLocation>file:///C:/Folder1/results.csv</dyn:fileLocation>
      <dyn:fileType>csv</dyn:fileType>
      <dyn:mimeType>application/csv</dyn:mimeType>
      <dyn:valueType>double</dyn:valueType>
      <dyn:numberOfHeaderLines>1</dyn:numberOfHeaderLines>
      <dyn:fieldSeparator>,</dyn:fieldSeparator>
      <dyn:decimalSymbol>.</dyn:decimalSymbol>
      <dyn:idColumnNo>1</dyn:idColumnNo>
      <dyn:idValue>wall1</dyn:idValue>
      <dyn:timeColumnNo>2</dyn:timeColumnNo>
      <dyn:valueColumnNo>6</dyn:valueColumnNo>
    </dyn:TabulatedFileTimeseries>
  </dyn:dynamicData>
</dyn:Dynamizer>
```

The Dynamizer defines the following parameters:

- **attributeRef** refers to the generic attribute *globalRadMonth* of the RoofSurface with the gml:id *wall1*.
- **startTime** and **endTime** represent the entire temporal extent for the dynamic values.

The actual parameters with details on the CSV file are specified by the Dynamizer using a TabulatedFileTimeseries object:

- **firstTimestamp** and **lastTimestamp** represent the temporal extent specified within the CSV file.
- **observationProperty** indicates the observed property (here: *GlobalIrradiationPerMonth*).
- **uom** specifies the unit of measurement of the observed property (here: *kWh*).
- **fileLocation** specifies the location where the file is stored.
- **fileType** indicates the file type (here: *csv*).
- `mimeType` specifies the MIME type associated with the file type (here: `application/csv`).
- `valueType` specifies the type of the values stored in the CSV file. The simulation results recorded in the CSV file are of type `double`.
- `numberOfHeaderLines` indicates the number of header lines in the CSV file (here: 1). In case there is no header line in the CSV file, it can be shown as 0.
- `fieldSeparator` represents the type of field separator used in the file (here: ',').
- `decimalSymbol` represents the decimal symbol used in the file (here: '.').
- `idColumnNo` indicates the column number that contains the IDs of the building surfaces.
- `timeColumnNo` indicates the column number that contains the timestamps of the irradiation values.
- `valueColumnNo` indicates the column number that contains the irradiation values.
- `idValue` indicates the value of the id used for querying. In this example, the simulation results are stored for different building surfaces such as `wall1`, `wall2`, `wall3`, etc. Since the `attributeRef` parameter refers to the RoofSurface `wall1`, the `idValue` parameter is set to `wall1`. It allows to define that only surfaces with the ID `wall1` are used from this TabulatedFileTimeseries object.

By specifying these parameters, the Dynamizer only retrieves irradiation values from the column `Global` where the timestamp provided in column `Time` is within the `startTime` and `endTime` defined by the Dynamizer, and where the ids in column `Surface_ID` are equal to the value `wall1`.

Instead of providing the column ids, it is also possible to specify the column names. This is illustrated in Listing 13.
Listing 13. Dynamizer using a TabulatedFileTimeseries object for referring to timeseries stored in an external CSV file using the column names.

```xml
<dyn:Dynamizer gml:id="global_irradiation_Dynamizer">
  <!-- attribute reference, start time and end time as in {Listing 12} -->
  <dyn:dynamicData>
    <dyn:TabulatedFileTimeseries>
      <dyn:firstTimestamp>2015-01-01T00:00:00Z</dyn:firstTimestamp>
      <dyn:lastTimestamp>2016-01-01T00:00:00Z</dyn:lastTimestamp>
      <dyn:observationProperty>GlobalIrradiationPerMonth</dyn:observationProperty>
      <dyn:uom>kWh</dyn:uom>
      <dyn:fileLocation>
        file:///C:/Folder1/results.csv
      </dyn:fileLocation>
      <dyn:fileType>csv</dyn:fileType>
      <dyn:mimeType>application/csv</dyn:mimeType>
      <dyn:valueType>double</dyn:valueType>
      <dyn:numberOfHeaderLines>1</dyn:numberOfHeaderLines>
      <dyn:fieldSeparator>,</dyn:fieldSeparator>
      <dyn:decimalSymbol>.</dyn:decimalSymbol>
      <dyn:idColumnName>Surface_ID</dyn:idColumnName>
      <dyn:idValue>wall1</dyn:idValue>
      <dyn:timeColumnName>Time</dyn:timeColumnName>
      <dyn:valueColumnName>Global</dyn:valueColumnName>
    </dyn:TabulatedFileTimeseries>
  </dyn:dynamicData>
</dyn:Dynamizer>
```

B.1.2.2. Example 2: Representation of timeseries data in files based on international standards

It is also possible to represent the timeseries values according to international standards such as OGC TimeseriesML 1.0 using the StandardFileTimeseries object.

An example is shown in Listing 14. The parameters used are identical to those described above for the TabulatedFileTimeseries object, except that the parameters referring to individual columns in the file are not required here. The monthly simulation results for the building wall surface are represented in a file results.xml according to the OGC TimeseriesML 1.0 standard; thus, the StandardFileTimeseries object references this file in the parameter `fileLocation` and `timeseriesml_1.0` is specified as file type and `application/xml` as MIME type.
Listing 14. Dynamizer using a GenericTimeseries object to represent timeseries inline.

```xml
<dyn:Dynamizer gml:id="global_irradiation_Dynamizer">
  <dyn:attributeRef>
    <!-- Single line without linebreak and space -->
    //con:WallSurface[@gml:id='wall1']/core:genericAttribute
    /gen:DoubleAttribute[gen:name='globalRadMonth']/gen:value
  </dyn:attributeRef>
  <dyn:startTime frame="#ISO-8601">2015-01-01T00:00:00Z</dyn:startTime>
  <dyn:endTime frame="#ISO-8601">2016-01-01T00:00:00Z</dyn:endTime>
  <dyn:dynamicData>
    <dyn:StandardFileTimeseries>
      <dyn:firstTimestamp>2015-01-01T00:00:00Z</dyn:firstTimestamp>
      <dyn:lastTimestamp>2016-01-01T00:00:00Z</dyn:lastTimestamp>
      <dyn:observationProperty>GlobalIrradiationPerMonth</dyn:observationProperty>
      <dyn:uom>kWh</dyn:uom>
      <dyn:fileLocation>
        file:///C:/Folder1/results.xml
      </dyn:fileLocation>
      <dyn:fileType>timeseriesml_1.0</dyn:fileType>
      <dyn:mimeType>application/xml</dyn:mimeType>
    </dyn:StandardFileTimeseries>
  </dyn:dynamicData>
</dyn:Dynamizer>
```

B.1.2.3. Example 3: Representation of timeseries data inline with the city object

The solar potential simulation results can also be represented inline using a GenericTimeseries object. It allows for defining the necessary metadata of the timeseries data as well as the timeseries data itself, as is shown in Listing 15.
Listing 15. Dynamizer using a GenericTimeseries object to represent timeseries inline.

```xml
<dyn:Dynamizer gml:id="global_irradiation_Dynamizer">
  <!-- attribute reference, start time and end time as in {Listing 12} -->
  <dyn:dynamicData>
    <dyn:GenericTimeseries>
      <dyn:firstTimestamp>2015-01-01T00:00:00Z</dyn:firstTimestamp>
      <dyn:lastTimestamp>2016-01-01T00:00:00Z</dyn:lastTimestamp>
      <dyn:observationProperty>GlobalIrradiationPerMonth</dyn:observationProperty>
      <dyn:uom>kWh</dyn:uom>
      <dyn:valueType>double</dyn:valueType>
      <dyn:timeValuePair>
        <dyn:TimeValuePair>
          <dyn:timestamp>2015-01</dyn:timestamp>
          <dyn:doubleValue>4293.446</dyn:doubleValue>
        </dyn:TimeValuePair>
      </dyn:timeValuePair>
      <dyn:timeValuePair>
        <dyn:TimeValuePair>
          <dyn:timestamp>2015-02</dyn:timestamp>
          <dyn:doubleValue>5563.502</dyn:doubleValue>
        </dyn:TimeValuePair>
      </dyn:timeValuePair>
      <!-- ... timeValuePair elements for the months March to December ... -->
    </dyn:GenericTimeseries>
  </dyn:dynamicData>
</dyn:Dynamizer>
```

B.1.2.4. Representing complex periodic patterns using Dynamizers

Dynamizers also support the composition of atomic timeseries, i.e. of GenericTimeseries objects, TabulatedFileTimeseries objects, and/or StandardFileTimeseries objects, to form composite timeseries. One typical example illustrating such a scenario is the mapping of energy values of a building for every hour in a day as is shown in Figure 14. The hourly consumption readings for a working day, a Saturday, and a Sunday are represented as individual atomic timeseries A, B, and C, respectively.
A CompositeTimeseries object may now contain five repetitions of timeseries A to reflect a pattern of energy consumption values for all weekdays, whereas the energy consumption for Saturday is represented by timeseries B and for Sunday by timeseries C. In this way, a weekly pattern can be defined containing the energy values for all seven days of a week (represented as 'AAAAAABC'). This is illustrated in [listing-gml-composite-timeseries].

1. Dynamizer defining a CompositeTimeseries to represent a weekly pattern of energy consumption values.

```xml
< dyn:Dynamizer gml:id="electricity_consumption">
  < dyn:attributeRef />
  < dyn:startTime >2015-W01-01</ dyn:startTime >
  < dyn:endTime >2015-W02-01</ dyn:endTime >
  < dyn:dynamicData >
    < dyn:CompositeTimeseries >
      < dyn:component >
        < dyn:TimeseriesComponent >
          < dyn:repetitions >5</ dyn:repetitions >
          < dyn:timeseries >
            < dyn:GenericTimeseries gml:id="Weekdays" />
            < dyn:firstTimestamp >T00:00:00</ dyn:firstTimestamp >
            < dyn:lastTimestamp >T24:00:00</ dyn:lastTimestamp >
            < dyn:observationProperty >ElecConsump</ dyn:observationProperty >
            < dyn:uom >kWh</ dyn:uom >
            < dyn:valueType >double</ dyn:valueType >
          </ dyn:timeseries >
        </ dyn:TimeseriesComponent >
      </ dyn:component >
    </ dyn:CompositeTimeseries >
  </ dyn:dynamicData >
</ dyn:Dynamizer >
```
<dyn:GenericTimeseries gml:id="Saturday">
    <!-- ISO 8601 Time Representation showing relative timestamps-->
    <dyn:timeseries>
        <dyn:timeValuePair>
            <dyn:TimeValuePair>
                <dyn:timestamp>T00:00:00</dyn:timestamp>
                <dyn:doubleValue>1.39</dyn:doubleValue>
            </dyn:TimeValuePair>
        </dyn:timeValuePair>
        <dyn:timeValuePair>
            <dyn:TimeValuePair>
                <dyn:timestamp>T01:00:00</dyn:timestamp>
                <dyn:doubleValue>1.44</dyn:doubleValue>
            </dyn:TimeValuePair>
        </dyn:timeValuePair>
        <dyn:timeValuePair>
            <dyn:TimeValuePair>
                <dyn:timestamp>T02:00:00</dyn:timestamp>
                <dyn:doubleValue>1.52</dyn:doubleValue>
            </dyn:TimeValuePair>
        </dyn:timeValuePair>
    </dyn:timeseries>
</dyn:GenericTimeseries>
The property `attributeRef` refers to a specific city object property or generic attribute to be overridden by the dynamic energy consumption values.

The properties `startTime` and `endTime` represent the overall temporal extent of the Dynamizer.
feature, which is one week in this example. The temporal extent is represented using the ISO 8601 Week Date Representation. This representation is defined using the string “YYYY-Www-D”, where [YYYY] is the year, [Www] is the week number prefixed by the letter W, from W01 through W53, and [D] is the weekday number, from 1 through 7, beginning with Monday and ending with Sunday. Therefore, the string “2015-W01-01” in this listing indicates Monday of the week number 1 of the year 2015. Similarly, the string “2015-W01-07” indicates Sunday of the week number 1 of the year 2015. The startTime and endTime follow the Half-Open interval, which means the startTime is inclusive and the endTime is exclusive.

The CompositeTimeseries object includes 3 components in this listing, representing weekdays, Saturdays, and Sundays, respectively. The component for weekdays includes a GenericTimeseries object indicating the electricity consumption values for every hour in a day. Within this GenericTimeseries object, the attributes firstTimestamp and lastTimestamp are relative timestamps and represent the temporal extent for each weekday, starting at 0 am and ending at 0 am the next day; thus, the duration of the GenericTimeseries object is 24 hours. By repeating the GenericTimeseries object five times, a total duration of five days is obtained. Each observation is represented using TimeValuePair elements where the timestamp is a time value relative to the absolute value defined within the temporal extent. For each iteration, the combination of this relative timestamp and the absolute timestamp give a complete timestamp including the date and time values. The property repetitions has the value 5 for this component to represent 5 iterations for all the weekdays in the week. In a similar way, the other components for Saturday and Sunday can be defined using the repetition value 1 for restriction to only 1 iteration.

![Figure 15. Example of a monthly pattern composed of individual CompositeTimeseries defined by complex energy consumption pattern composed of atomic timeseries (source: Chaturvedi 2021).](image_url)

[listing-gml-composite-timeseries] is an example for a CompositeTimeseries object representing periodic patterns of electricity consumption values for one week. Depending on the use case, more complex CompositeTimeseries objects can be defined based on individual CompositeTimeseries components, such as a monthly pattern consisting of 5 weekly patterns as is shown in Figure 15 or a yearly pattern which consists of 52 repetitions of the weekly pattern. Similarly, a CompositeTimeseries object could contain components for weekdays only to represent weekday patterns.
B.2. Point Cloud Examples

The PointCloud module allows for representing the geometries of city objects by 3D point clouds in three different options, which are illustrated in the following.

The CityGML and LAZ data sets presented here are available under https://github.com/opengeospatial/CityGML-3.0Encodings/tree/master/CityGML/Examples/PointCloud/Real-world%20examples/CityGML_3.0_buildings_with_point_cloud_representation.

**Option 1: Inline representation**

A MultiPoint geometry is provided inline with the CityGML file, i.e., the point cloud of a specific feature is embedded directly within the feature represented by the point cloud. Listing 16 shows an example for a point cloud of a building represented inline the Building feature.

Figure 16 shows different visualisations of the GML file. The left image displays the building features themselves, the middle image shows the point clouds of the buildings, and the right image is a combination of the building features and the point clouds.

![Figure 16. Visualization of the inline representation of points clouds.](image-url)
Listing 16. The point cloud of a building is embedded in the Building feature.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
  <core:cityObjectMember>
    <bldg:Building gml:id="DEBY_LOD2_4903217">
      <gml:name>DEBY_LOD2_4903217</gml:name>
      <core:creationDate>2015-09-24T00:00:00</core:creationDate>
      <core:boundary> ... </core:boundary>
      ...
      <core:pointCloud>
        <pcl:PointCloud gml:id="93648526a0d3">
          <pcl:points>
            <gml:MultiPoint gml:id="93648526a0d3-0" srsName="urn:adv:crs:DE_DHDN_3GK4*DE_DHN92_NH" srsDimension="3">
              <gml:pointMember>
                <gml:Point gml:id="93648526a0d3-1">
                  <gml:pos>4467308.6 5331553.4 539.28</gml:pos>
                </gml:Point>
              </gml:pointMember>
              <gml:pointMember>
                <gml:Point gml:id="93648526a0d3-2">
                  <gml:pos>4467308.6 5331553 539.62</gml:pos>
                </gml:Point>
              </gml:pointMember>
              <gml:pointMember>
                <gml:Point gml:id="93648526a0d3-3">
                  <gml:pos>4467309 5331553 539.66</gml:pos>
                </gml:Point>
              </gml:pointMember>
              ... further points ...
            </gml:MultiPoint>
          </pcl:points>
        </pcl:PointCloud>
      </core:pointCloud>
      <con:height>
        <con:Height>
          <con:highReference>topOfConstruction</con:highReference>
          <con:lowReference>lowestGroundPoint</con:lowReference>
          <con:status>measured</con:status>
          <con:value uom="urn:adv:uom:m">24.709</con:value>
        </con:Height>
      </con:height>
      <bldg:function>99999_1001</bldg:function>
      <bldg:roofType>3100</bldg:roofType>
    </bldg:Building>
  </core:cityObjectMember>
</core:CityModel>
```
Option 2: External representation in multiple files
A separate point cloud file is provided for every feature. Each feature in the CityGML file references
the corresponding point cloud file. This is shown in Listing 17, where the building feature
references the LAZ file DEBY_LOD2_4903217.laz that contains the points for this building
(illustrated in Figure 17).
Listing 17. The point cloud of a building is referenced in an external file.

<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
<core:cityObjectMember>
<bldg:Building gml:id="DEBY_LOD2_4903217">
<gml:name>DEBY_LOD2_4903217</gml:name>
<core:creationDate>2015-09-24T00:00:00</core:creationDate>
<core:boundary> ... </core:boundary>
...
<core:pointCloud>
<pcl:PointCloud gml:id="DEBY_LOD2_4903217_c0e8322c-8cf4-4fdd-8ddc4780566814a5">
<pcl:pointFile>..\pointcloud\DEBY_LOD2_4903217.laz</pcl:pointFile>
</pcl:PointCloud>
</core:pointCloud>
<con:height>
<con:Height>
<con:highReference>topOfConstruction</con:highReference>
<con:lowReference>lowestGroundPoint</con:lowReference>
<con:status>measured</con:status>
<con:value uom="urn:adv:uom:m">24.709</con:value>
</con:Height>
</con:height>
<bldg:function>99999_1001</bldg:function>
<bldg:roofType>3100</bldg:roofType>
</bldg:Building>
</core:cityObjectMember>
</core:CityModel>

Figure 17. Visualization of the externally referenced LAZ file containing the point cloud of one building.

Option 3: External representation in a single file
One point cloud file is provided that contains all points from all features in a specific area. In the
point cloud file, all points belonging to a specific feature have the same value set in the attribute
“point_source_id”. Each feature in the CityGML instance document references the point cloud file

106


and all points with the corresponding value in the attribute “point_source_id”. In Listing 18, the building feature references all points with the “point_source_id” value 132 in the LAZ file 4467_5331_40_bDOM_classified.laz. The LAZ file itself is shown in Figure 18. Highlighted are those points with the “point_source_id” value 132.

Listing 18. The point cloud of a building is referenced in an external file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
  <core:cityObjectMember>
    <bldg:Building gml:id="DEBY_LOD2_4903217">
      <gml:name>DEBY_LOD2_4903217</gml:name>
      <core:creationDate>2015-09-24T00:00:00</core:creationDate>
      ... </core:boundary>
    </bldg:Building>
  </core:cityObjectMember>
</core:CityModel>
```
B.3. Versioning Examples

Version Management using XLinks

The CityGML 3.0 Conceptual Model introduces the concept of versioned features. The individual versions of a specific city object can be identified by their featureID attribute, which is different for each version of the same real-world object. The featureID attribute is mapped to the gml:id attribute in CityGML 3.0 data sets encoded in GML. This gml:id is used for referencing versioned features. See Section 6.1.2 for more information on identifiers for features.

To ensure that new versions of a city object are always created and referenced in the same way, the following rules have been defined for version management with CityGML 3.0. The rules clearly define when to create new versions of features and when to reference existing versions of features. The relationship between versioned top-level features and their corresponding versioned subordinate features is always unambiguous, since the versioned subordinate features are clearly assigned to their corresponding versioned top-level features. In this way, it is also possible to see directly for each top-level feature, whether the top-level feature itself or its subordinate features have changed. This also makes database queries very easy.

**Rule 1: Only the top-level feature changes**

1. When only the top-level feature changes, a new version of the top-level feature shall be created which receives a new gml:id.

2. All subordinate features, i.e. the features that are part of the top-level feature, must remain unchanged and must be referenced in their unchanged version by the new version of the top-level feature via their gml:id using XLink.

With this rule, the subordinate features can remain in their current state and do not need to be copied as new versions for every change within in a top-level feature.

The use of Rule 1 is illustrated in Figure 19. The function attribute of the top-level feature Building changes from Living to Office; thus, a new version of the Building feature is created with a new
Since the subordinate feature BuildingPart remains unchanged, the new Building version references the unchanged BuildingPart feature.

![Diagram of version management](image)

**Figure 19.** Version management when only the top-level feature changes.

**Rule 2: Only a subordinate feature changes**

1. When only a subordinate feature changes, a new version of this subordinate feature shall be created which receives a new gml:id.

2. This subordinate feature is referenced by a superordinate feature. By creating a new version of the subordinate feature with a new gml:id, the XLink reference of the superordinate to the subordinate feature must be adapted to this new gml:id as well, which means that also a new version of the superordinate feature must be created.

3. Changes to these XLink references and creation of new superordinate features must be applied transitively to all superordinate features up to the top-level feature.

4. All other subordinate features that do not change must be referenced in their unchanged versions by the new versions of the superordinate features according to Rule 1.

For aggregation hierarchies this means that whenever a subordinate feature changes, also new versions of all corresponding superordinate features up to the top-level feature need to be created.

The use of rule 2 is illustrated in Figure 20. Here the roof type of the subordinate feature BuildingPart changes from *Flat* to *Saddle* which results in a new version of the BuildingPart feature with a new gml:id. The feature is referenced from a superordinate feature, i.e., the top-level feature Building. This means that also a new version needs to be created for the Building feature with a new gml:id and a reference to the new BuildingPart feature.
Figure 20. Version management when only a subordinate feature changes that is directly referenced from the top-level feature.

Figure 21 illustrates rule 2 for an aggregation hierarchy. Here, the BuildingPart feature has two subordinate features BuildingRoom. The function of one BuildingRoom changes from *Bedroom* to *Home Office*, whereas the other BuildingRoom feature remains unchanged. This means that a new version is created for the changed BuildingRoom feature and also for the superordinate BuildingPart feature. The new BuildingPart feature references then the changed BuildingRoom feature and the unchanged BuildingRoom feature. Since BuildingPart, in turn, is subordinate to the top-level feature Building, also a new version of the Building feature is created.
Rule 3: Top-level and subordinate features change

1. This rule is a combination of Rules 1 and 2; i.e., when a top-level feature and its corresponding subordinate features change, both rules must be applied jointly.
The use of rule 3 is illustrated in Figure 20. Here, the top-level feature Building and its subordinate feature BuildingPart change. Thus, new versions are created for both features with new gml:ids, new attribute values and a reference from the new Building feature to the new BuildingPart feature.

**Example: Successive version management over a period of time**

The example in Figure 23 illustrates different versions of a building at three points in time. The original Building including its BuildingPart exists at time t1. At time t2, the top-level feature Building changes its function which results in the application of Rule 1. Afterwards at time t3, the subordinate feature BuildingPart changes its roof type, thus, Rule 2 is applied in this case. Listing 19 represents this example in GML.

![Figure 23. Successive version management over a period of time.](image)
Listing 19. GML instance document for the example on successive version management over a period of time illustrated in Figure 23.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
  <core:cityObjectMember>
    <bldg:Building gml:id="B1020_t1">
      <gml:identifier codeSpace="www.test.org/xyz">B1020</gml:identifier>
      <core:creationDate>2012-08-02T00:00:00</core:creationDate>
      <core:terminationDate>2013-10-09T00:00:00</core:terminationDate>
      <bldg:function>Office</bldg:function>
      <bldg:buildingPart>
        <bldg:BuildingPart gml:id="BP12_t1">
          <gml:identifier codeSpace="www.test.org/xyz">BP12</gml:identifier>
          <core:creationDate>2012-08-02T00:00:00</core:creationDate>
          <core:terminationDate>2014-06-03T00:00:00</core:terminationDate>
          <bldg:roofType>Flat</bldg:roofType>
        </bldg:BuildingPart>
      </bldg:buildingPart>
    </bldg:Building>
  </core:cityObjectMember>
  <core:cityObjectMember>
    <bldg:Building gml:id="B1020_t2">
      <gml:identifier codeSpace="www.test.org/xyz">B1020</gml:identifier>
      <core:creationDate>2013-10-09T00:00:00</core:creationDate>
      <bldg:function>Living</bldg:function>
      <bldg:buildingPart xlink:href="#BP12_t1"/>
    </bldg:Building>
  </core:cityObjectMember>
  <core:cityObjectMember>
    <bldg:Building gml:id="B1020_t3">
      <gml:identifier codeSpace="www.test.org/xyz">B1020</gml:identifier>
      <core:creationDate>2014-06-03T00:00:00</core:creationDate>
      <bldg:function>Living</bldg:function>
      <bldg:buildingPart>
        <bldg:BuildingPart gml:id="BP12_t3">
          <gml:identifier codeSpace="www.test.org/xyz">BP12</gml:identifier>
          <core:creationDate>2014-06-03T00:00:00</core:creationDate>
          <bldg:roofType>Saddle</bldg:roofType>
        </bldg:BuildingPart>
      </bldg:buildingPart>
    </bldg:Building>
  </core:cityObjectMember>
</core:CityModel>
```

Example: Successive version management with Versions and VersionTransitions

The example in Figure 24 illustrates 4 Building features (b1, b2-1, b2-2, and b3) represented across 2 Versions (v1 and v2). A historicalSuccession type VersionTransition (vt1) stores the changes between each Building through 3 Transactions (t1, t2, and t3). t1 shows that Building b1 is deleted from v1, t2
shows that Building b2-1 is replaced by Building b2-2 in v2, and t3 shows that Building b3 is inserted into v2. Listing 20 represents this example in GML.

Listing 20. GML instance document for the example on successive version management using Versions and VersionTransitions illustrated in Figure 24.

```
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
  <core:versionMember>
    <vers:Version gml:id="v1">
      <vers:tag>v1.0</vers:tag>
      <vers:versionMember xlink:href="#b1"/>
      <vers:versionMember xlink:href="#b2-1"/>
    </vers:Version>
  </core:versionMember>
  <core:versionMember>
    <vers:Version gml:id="v2">
      <vers:tag>v2.0</vers:tag>
      <vers:versionMember xlink:href="#b2-2"/>
      <vers:versionMember xlink:href="#b3"/>
    </vers:Version>
  </core:versionMember>
</core:CityModel>
```
To exemplify a version transition between two city snapshots:

- **Delete**: 
  - Old Feature: b1
  - New Feature: b2-1

- **Replace**: 
  - Old Feature: b1
  - New Feature: b2-2

- **Insert**: 
  - New Feature: b3

**Core City Object Members**:

- Building b1
- Building b2-1
- Building b2-2
- Building b3
B.4. Address Examples

The CityGML 3.0 CM provides the ability to represent address information of real-world features. Addresses can be declared for AbstractBridge, AbstractBuilding, BuildingUnit, Door, and DoorSurface feature types using their respective address properties and the Address feature type of the Core module. An Address can be declared inline or by an XLink reference.

An Address is modeled with two properties:

- An optional multiPoint property containing a gml:MultiPoint for specifying the 2D or 3D position of the address.
- A xalAddress property containing a xAL:Address type from the OASIS CIQ TC extensible Address Language standard (xAL).

The CityGML datasets presented here are available under https://github.com/opengeospatial/CityGML-3.0Encodings/tree/master/CityGML/Examples/Building, additional examples are also available on the OASIS documentation repository http://docs.oasis-open.org/ciq/v3.0/cs02/xsd/default/examples/xAL/.

Example 1: Structured address and multipoint

xAL provides a rich vocabulary for structuring addresses in GML. Listing 21 shows how an address in Germany can be modeled using xAL and a multipoint:
Example 2: Free text address

More freeform addresses can also be declared using FreeTextAddresses as shown in Listing 22:
B.5. Application Domain Extension (ADE) Examples

The CityGML 3.0 Conceptual Model introduces the concept of the Application Domain Extension (ADE) to ensure extensibility of the CityGML data model as in the previous version CityGML 2.0. The ADE mechanism allows to add new application-specific feature types, to inject additional properties to the predefined CityGML feature types, to associate application data with predefined CityGML content, or to define value domains for attributes.

Since CityGML 3.0 applies the Model-Driven Architecture approach which separates the conceptual model and the encoding, the development of ADEs differs from CityGML 2.0. In a first step, ADEs shall be defined as UML conceptual models. Afterwards, XML schemas shall be derived automatically from the UML conceptual models using suitable software tools. This is identical to the creation of the CityGML 3.0 XML schemas which have been derived automatically from the CityGML 3.0 UML Conceptual Model.

This chapter will illustrate the workflow of ADE generation using the example of two ADEs: 1) a so-called Test ADE, which is an artificial ADE that comprises most possible scenarios that can occur in an ADE, and 2) the Urban Planning ADE as real-world example.
For both ADEs, UML models have been created based on the rules described in Chapter 9 of the CityGML 3.0 Conceptual Model standard for how to define UML models for ADEs. The UML models were created using the Enterprise Architect project file that already contains the CityGML 3.0 UML model. Afterwards, the software tool ShapeChange was applied to the UML models to derive the XML schemas. Finally, test data sets were implemented that validate against the XML schemas.

**B.5.1. Example 1: Test ADE**

The Test ADE is a light-weight artificial ADE, which was designed for demonstrating the most frequent scenarios in practical ADEs. In particular, the both methods ('hook' mechanism and subclassing) for augmenting predefined CityGML feature classes with additional ADE Properties are presented. The UML Conceptual Model of the Test ADE is shown in Figure 25. All ADE classes can be categorized into the following groups:

1. The `BuildingUndergroundClassValue`, `BuildingUndergroundFunctionValue`, and `BuildingUndergroundUsageValue` show how to define ADE class attributes using code list.
2. The `EnergyPerformanceCertification` is an example of how to define an ADE data type, which can comprise a set of simple attributes.
3. The `BuildingProperties` is a subclass of the 'hook' data type `ADEOfAbstractBuilding` for augmenting the predefined CityGML class `AbstractBuilding` with additional properties.
4. The `Facility` and its subclasses `ElectricalAppliance` and `LightingFacility` are object types, which are not derived from any predefined CityGML classes.
5. The `IndustrialBuilding` and `MovingConstruction` are two top-level classes derived from the predefined CityGML top-level classes `Building` and `OtherConstruction` respectively.
6. The `IndustrialBuildingPart` and `SolarRoofSurface` are two example non-top-level classes, which are also derived from the CityGML thematic modules `Building` and `Construction`.
7. The `AbstractBuildingUnderground` and its subclasses `BuildingUnderground` and `BuildingUndergroundPart` are derived from the CityGML Core Module, and comprise a couple of new thematic and spatial properties in addition to the superclass `AbstractOccupiedSpace`.

A corresponding XML schema derived from the UML Conceptual Model is shown in Listing 23. An example ADE dataset is represented in Listing 24.
Figure 25. UML model of the Test ADE.

Listing 23. XML schema of the Test ADE model in Figure 25.

```xml
<?xml version="1.0" encoding="UTF-8"?><schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:bldg="http://www.opengis.net/citygml/building/3.0"
xmlns:con="http://www.opengis.net/citygml/construction/3.0"
xmlns:core="http://www.opengis.net/citygml/3.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:test="http://www.citygml.org/ade/TestADE/2.0" elementFormDefault="qualified"
targetNamespace="http://www.citygml.org/ade/TestADE/2.0" version="2.0">
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="../schema/core.xsd"/>
</schema>
```
<element abstract="true" name="AbstractBuildingUnderground" substitutionGroup="core:AbstractOccupiedSpace" type="test:AbstractBuildingUndergroundType"/>
<complexType abstract="true" name="AbstractBuildingUndergroundType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="energyPerformanceCertification" type="test:EnergyPerformanceCertificationPropertyType"/>
        <element minOccurs="0" name="lod0GenericGeometry" type="gml:GeometryPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="address" type="core:AddressPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="equippedWith">
          <complexType>
            <complexContent>
              <extension base="gml:AbstractMemberType">
                <sequence minOccurs="0">
                  <element ref="test:Facility"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType"/>
      </sequence>
      <!-- extension -->
    </extension>
  </complexContent>
</complexType>
<complexType name="AbstractBuildingUndergroundPropertyType">
  <sequence minOccurs="0">
    <element ref="test:AbstractBuildingUnderground"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="BuildingProperties" substitutionGroup="bldg:ADEOfAbstractBuilding" type="test:BuildingPropertiesType"/>
<complexType name="BuildingPropertiesType">
  <complexContent>
    <extension base="bldg:ADEOfAbstractBuildingType">
      <sequence>
        <element minOccurs="0" name="ownerName" type="string"/>
        <element name="energyPerformanceCertification" type="test:EnergyPerformanceCertificationPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="buildingUnderground">
          <complexType>
            <complexContent>
              <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                  <element ref="test:BuildingUnderground"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element minOccurs="0" name="floorArea" type="gml:AreaType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BuildingPropertiesPropertyType">
  <sequence>
    <element ref="test:BuildingProperties"/>
  </sequence>
</complexType>
<element name="BuildingUnderground" substitutionGroup="test:AbstractBuildingUnderground" type="test:BuildingUndergroundType"/>
<complexType name="BuildingUndergroundType">
  <complexContent>
    <extension base="test:AbstractBuildingUndergroundType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="consistsOf">
          <complexType>
            <complexContent>
              <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                  <element ref="test:BuildingUndergroundPart"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BuildingUndergroundPropertyType">
  <sequence minOccurs="0">
    <element ref="test:BuildingUnderground"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="BuildingUndergroundPart" substitutionGroup="test:AbstractBuildingUnderground" type="test:BuildingUndergroundPartType"/>
<complexType name="BuildingUndergroundPartType">
  <complexContent>
    <extension base="test:AbstractBuildingUndergroundType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
<complexType name="BuildingUndergroundPartPropertyType">
  <sequence minOccurs="0">
    <element ref="test:BuildingUndergroundPart"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="ElectricalAppliance" substitutionGroup="test:Facility" type="test:ElectricalApplianceType"/>
<complexType name="ElectricalApplianceType">
  <complexContent>
    <extension base="test:FacilityType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
<complexType name="ElectricalAppliancePropertyType">
  <sequence minOccurs="0">
    <element ref="test:ElectricalAppliance"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="EnergyPerformanceCertification" substitutionGroup="gml:AbstractObject" type="test:EnergyPerformanceCertificationType"/>
<complexType name="EnergyPerformanceCertificationType">
  <sequence>
    <element maxOccurs="unbounded" name="certificationName" type="string"/>
    <element name="certificationId" type="string"/>
  </sequence>
</complexType>
<complexType name="EnergyPerformanceCertificationPropertyType"/>
<element ref="test:EnergyPerformanceCertification"/>
</element>
</complexType>
<element name="Facility" substitutionGroup="gml:AbstractGML" type="test:FacilityType"/>
<complexType name="FacilityType">
<complexContent>
<extension base="gml:AbstractGMLType">
<sequence>
<element name="electricalPower" type="gml:MeasureType"/>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="FacilityPropertyType">
<sequence minOccurs="0">
<element ref="test:Facility"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="IndustrialBuilding" substitutionGroup="bldg:Building" type="test:IndustrialBuildingType"/>
<complexType name="IndustrialBuildingType">
<complexContent>
<extension base="bldg:BuildingType">
<sequence>
<element minOccurs="0" name="remark" type="string"/>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="IndustrialBuildingPropertyType">
<sequence minOccurs="0">
<element ref="test:IndustrialBuilding"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="IndustrialBuildingPart" substitutionGroup="bldg:BuildingPart" type="test:IndustrialBuildingPartType"/>
<complexType name="IndustrialBuildingPartType">
<complexContent>
<extension base="bldg:BuildingPartType">
<sequence>
<element minOccurs="0" name="remark" type="string"/>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="IndustrialBuildingPartPropertyType">
  <sequence minOccurs="0">
    <element ref="test:IndustrialBuildingPart"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="LightingFacility" substitutionGroup="test:Facility" type="test:LightingFacilityType"/>

<complexType name="LightingFacilityType">
  <complexContent>
    <extension base="test:FacilityType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>

<complexType name="LightingFacilityPropertyType">
  <sequence minOccurs="0">
    <element ref="test:LightingFacility"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="MovingConstruction" substitutionGroup="con:OtherConstruction" type="test:MovingConstructionType"/>

<complexType name="MovingConstructionType">
  <complexContent>
    <extension base="con:OtherConstructionType">
      <sequence>
        <element minOccurs="0" name="remark" type="string"/>
        <element maxOccurs="unbounded" minOccurs="0" name="coveredBy">
          <complexType>
            <complexContent>
              <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                  <element ref="test:SolarRoofSurface"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="MovingConstructionPropertyType">
  <sequence minOccurs="0">
    <element ref="test:MovingConstruction"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
Listing 24. Example dataset implementing the illustrated Test ADE schema.

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CityModel xmlns:con="http://www.opengis.net/citygml/construction/3.0"
xmlns:tran="http://www.opengis.net/citygml/transportation/3.0"
xmlns:wtr="http://www.opengis.net/citygml/waterbody/3.0"
xmlns:veg="http://www.opengis.net/citygml/vegetation/3.0"
xmlns="http://www.opengis.net/citygml/3.0"
xmlns:dem="http://www.opengis.net/citygml/relief/3.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:bldg="http://www.opengis.net/citygml/building/3.0"
xmlns:ct="urn:oasis:names:tc:ciq:ct:3"
xmlns:grp="http://www.opengis.net/citygml/cityobjectgroup/3.0"
xmlns:dyn="http://www.opengis.net/citygml/dynamizer/3.0"
xmlns:pnt="http://www.opengis.net/citygml/pointcloud/3.0"
xmlns:tun="http://www.opengis.net/citygml/tunnel/3.0"
xmlns:frn="http://www.opengis.net/citygml/cityfurniture/3.0"
xmlns:gen="http://www.opengis.net/citygml/generics/3.0"
xmlns:xAL="urn:oasis:names:tc:ciq:xal:3"
xmlns:app="http://www.opengis.net/citygml/appearance/3.0"
xmlns:luse="http://www.opengis.net/citygml/landuse/3.0"
xmlns:brid="http://www.opengis.net/citygml/bridge/3.0"
xmlns:ver="http://www.opengis.net/citygml/versioning/3.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:test="http://www.citygml.org/ade/TestADE/2.0">
<cityObjectMember>
<test:IndustrialBuilding>
  <gml:description>FZK/IAI test cases only</gml:description>
  <gml:name>Testcase-0-0_LoD2</gml:name>
</test:IndustrialBuilding>
</cityObjectMember>
B.5.2. Example 2: Urban Planning ADE

In this example, ADE generation and data implementation of the Urban Planning ADE for CityGML 3.0 are explained. This is based on actual ADE schemas and data of the Urban Planning ADE with exist already for CityGML 2.0 and are published as Open Standard "Data Encoding Specification of i-Urban Revitalization -Urban Planning ADE-" and Open Data from the Project PLATEAU in Japan.

The Urban Planning ADE has been published to realize "i-Urban Revitalization", an information infrastructure dedicated for urban planning to contribute to data-driven urban development and urban revitalization. In the Urban Planning ADE, four modules are defined: Urban Object, Urban Function, Statistical Grid Module, and Public Transit. See OGC Discussion Paper (OGC 20-000r1) and ISPRS annals for more information about the Urban Planning ADE.
Step 1: Designing the ADE as UML conceptual model

This example shows how to convert the Urban Object (namespace: uro) module from CityGML 2.0 to CityGML 3.0 and how to inject additional properties to the predefined CityGML Building feature type. In order to allow to use multiple distinct ADEs for the same CityGML feature type, the CityGML 3.0 specification recommends to inject additional properties to a predefined feature type by a mechanism called "hook" rather than defining a subclass that inherits from a predefined CityGML feature type.

The hook has been implemented in the the CityGML 3.0 Conceptual Model as data type _ADEOf<FeatureTypeName>_ defined for each feature type where <FeatureTypeName> is the name of that feature type. The UML models for the Urban Object module has been implemented as described in the Requirement 50 in section 9.5 in CityGML 3.0 Conceptual Model. The UML model has been defined using the software tool Enterprise Architect and is available here as .eap file.

The UML conceptual model is illustrated in Figure 26. The DataType BuildingProperties is modelled as subclass of the "hook" data type _ADEOfAbstractBuilding_ and defines the additional properties that are added to the CityGML AbstractBuilding class. Additionally, a DataType KeyValuePair and a Union ValueType are defined. By using the BuildingProperties that includes two sets of properties, BuildingDetails and LargeCustomerFacilities, 3D city models can systematically be extended with the detailed properties for urban planning.

Step 2: Deriving the XML schema from the UML model
After the UML model has been defined, an XML schema can be derived from the UML model automatically using ShapeChange. To be able to do so, first a configuration file needs to be developed which ShapeChange needs for executing the conversion in a correct way.

The command below shows how to convert the UML model to an XML schema by running ShapeChange via a command shell:

```
java -jar ShapeChange-2.11.0.jar -Dfile.encoding=UTF-8 -c ShapeChangeConfigurationFile_CityGML_3.0_UrbanPlanningADE.xml
```

The ShapeChange configuration file is illustrated in Listing 25 and the derived XML schema from the Enterprise Architect UML model is illustrated in Listing 26.

Listing 25. ShapeChange configuration file used for converting the ADE UML model to an XML schema.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ShapeChangeConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xi="http://www.w3.org/2001/XInclude"
    xmlns="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1"
    xmlns:sc="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1"
    xsi:schemaLocation="http://www.interactive-instruments.de/ShapeChange/Configuration/1.1
    http://shapechange.net/resources/schema/ShapeChangeConfiguration.xsd">
    <input>
        <parameter name="inputModelType" value="EA7"/>
        <parameter name="inputFile" value="CityGML_3.0_UrbanPlanningADE.eap"/>
        <parameter name="appSchemaName" value="UrbanObjectData"/>
        <parameter name="checkingConstraints" value="disabled"/>
        <parameter name="addTaggedValues" value="itemType"/>
        <xi:include href="http://shapechange.net/resources/config/StandardAliases.xml"/>
        <stereotypeAliases>
            <StereotypeAlias wellknown="property" alias="Property"/>
            <StereotypeAlias wellknown="version" alias="Version"/>
            <StereotypeAlias wellknown="FeatureType" alias="TopLevelFeatureType"/>
        </stereotypeAliases>
    </input>
    
    <log>
        <parameter name="reportLevel" value="INFO"/>
        <parameter name="logFile" value="Log_CityGML_3.0_UrbanPlanningADE.xml"/>
    </log>
    
    <targets>
        <TargetXmlSchema
            class="de.interactive_instruments.ShapeChange.Target.XmlSchema.XmlSchema"
            mode="enabled">
            <targetParameter name="outputDirectory" value="."/>
            <targetParameter name="defaultEncodingRule" value="citygml"/>
        </TargetXmlSchema>
    </targets>
</ShapeChangeConfiguration>
```
<EncodingRule name="citygml" extends="iso19136_2007">
  <rule name="rule-xsd-rel-association-classes"/>
  <rule name="rule-xsd-clas-basictype"/>
  <rule name="rule-xsd-prop-initialValue"/>
  <rule name="rule-xsd-prop-att-map-entry"/>
  <rule name="rule-xsd-prop-constrainingFacets"/>
  <rule name="rule-xsd-all-notEncoded"/>
  <rule name="rule-xsd-clas-union-asGroup"/>
</EncodingRule>

<xi:include href="http://shapechange.net/resources/config/StandardRules.xml"/>
<xi:include href="http://shapechange.net/resources/config/StandardNamespaces.xml"/>
<xmlNamespaces>
  <XmlNamespace nsabr="app" ns="http://www.opengis.net/citygml/appearance/3.0" location="../schema/appearance.xsd"/>
  <XmlNamespace nsabr="brid" ns="http://www.opengis.net/citygml/bridge/3.0" location="../schema/bridge.xsd"/>
  <XmlNamespace nsabr="bldg" ns="http://www.opengis.net/citygml/building/3.0" location="../schema/building.xsd"/>
  <XmlNamespace nsabr="pcl" ns="http://www.opengis.net/citygml/pointcloud/3.0" location="../schema/pointCloud.xsd"/>
  <XmlNamespace nsabr="frn" ns="http://www.opengis.net/citygml/cityfurniture/3.0" location="../schema/cityFurniture.xsd"/>
  <XmlNamespace nsabr="grp" ns="http://www.opengis.net/citygml/cityobjectgroup/3.0" location="../schema/cityObjectGroup.xsd"/>
  <XmlNamespace nsabr="con" ns="http://www.opengis.net/citygml/construction/3.0" location="../schema/construction.xsd"/>
  <XmlNamespace nsabr="core" ns="http://www.opengis.net/citygml/3.0" location="../schema/core.xsd"/>
  <XmlNamespace nsabr="dyn" ns="http://www.opengis.net/citygml/dynamizer/3.0" location="../schema/dynamizer.xsd"/>
  <XmlNamespace nsabr="gen" ns="http://www.opengis.net/citygml/generics/3.0" location="../schema/generics.xsd"/>
  <XmlNamespace nsabr="luse" ns="http://www.opengis.net/citygml/landuse/3.0" location="../schema/landUse.xsd"/>
  <XmlNamespace nsabr="dem" ns="http://www.opengis.net/citygml/relief/3.0" location="../schema/relief.xsd"/>
  <XmlNamespace nsabr="tran" ns="http://www.opengis.net/citygml/transportation/3.0" location="../schema/transportation.xsd"/>
  <XmlNamespace nsabr="tun" ns="http://www.opengis.net/citygml/tunnel/3.0" location="../schema/tunnel.xsd"/>
  <XmlNamespace nsabr="veg" ns="http://www.opengis.net/citygml/vegetation/3.0" location="../schema/vegetation.xsd"/>
  <XmlNamespace nsabr="vers" ns="http://www.opengis.net/citygml/versioning/3.0" location="../schema/versioning.xsd"/>
  <XmlNamespace nsabr="wtr" ns="http://www.opengis.net/citygml/waterbody/3.0" location="../schema/waterBody.xsd"/>
  <XmlNamespace nsabr="xAL" ns="urn:oasis:names:tc:ciq:xal:3" location=""/>
</xmlNamespaces>
Listing 26. XML schema document of the ADE derived from UML model.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:bldg="http://www.opengis.net/citygml/building/3.0" 
    xmlns:uro="https://www.chisou.go.jp/tiiki/toshisaisei/itoshisaisei/iur/uro/1.5"
    elementFormDefault="qualified"
    targetNamespace="http://www.kantei.go.jp/jp/singi/tiiki/toshisaisei/itoshisaisei/iur/u" 
    ro/1.4" version="1.4.1">
    <import namespace="http://www.opengis.net/citygml/building/3.0"
        schemaLocation="../schema/building.xsd"/>

    <!--XML Schema document created by ShapeChange - http://shapechange.net/-->  
    <element name="BuildingDetails" substitutionGroup="gml:AbstractObject"
        type="uro:BuildingDetailsType">
        <complexType name="BuildingDetailsType">
            <sequence>
                <element minOccurs="0" name="serialNumberOfBuildingCertification"
                    type="string"/>
                <element minOccurs="0" name="siteArea" type="gml:MeasureType"/>
                <element minOccurs="0" name="buildingFootprintArea" type="gml:MeasureType"/>
                <element minOccurs="0" name="buildingRoofEdgeArea" type="gml:MeasureType"/>
                <element minOccurs="0" name="developmentArea" type="gml:MeasureType"/>
                <element minOccurs="0" name="totalFloorArea" type="gml:MeasureType"/>
                <element minOccurs="0" name="buildingStructureType" type="gml:CodeType"/>
                <element minOccurs="0" name="fireproofStructureType" type="gml:CodeType"/>
                <element minOccurs="0" name="implementingBody" type="string"/>
                <element minOccurs="0" name="urbanPlanType" type="gml:CodeType"/>
                <element minOccurs="0" name="districtsAndZonesType" type="gml:CodeType"/>
                <element minOccurs="0" name="landUsePlanType" type="gml:CodeType"/>
                <element minOccurs="0" name="areaClassificationType" type="gml:CodeType"/>
                <element minOccurs="0" name="prefecture" type="gml:CodeType"/>
                <element minOccurs="0" name="city" type="gml:CodeType"/>
                <element minOccurs="0" name="reference" type="string"/>
            </sequence>
        </complexType>
    </element>
</schema>
```
<element minOccurs="0" name="note" type="string"/>
<element minOccurs="0" name="surveyYear" type="gYear"/>
</sequence>
</complexType>
<complexType name="BuildingDetailsPropertyType">
  <sequence>
    <element ref="uro:BuildingDetails"/>
  </sequence>
</complexType>
<element name="BuildingProperties" substitutionGroup="bldg:ADEOfAbstractBuilding" type="uro:BuildingPropertiesType"/>
<complexType name="BuildingPropertiesType">
  <complexContent>
    <extension base="bldg:ADEOfAbstractBuildingType">
      <sequence>
        <element minOccurs="0" name="buildingDetails" type="uro:BuildingDetailsPropertyType"/>
        <element minOccurs="0" name="largeCustomerFacilities" type="uro:LargeCustomerFacilitiesPropertyType"/>
        <element maxOccurs="unbounded" minOccurs="0" name="extendedAttribute" type="uro:KeyValuePairPropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BuildingPropertiesPropertyType">
  <sequence>
    <element ref="uro:BuildingProperties"/>
  </sequence>
</complexType>
<element name="CityObjectGroupProperties" substitutionGroup="grp:ADEOfCityObjectGroup" type="uro:CityObjectGroupPropertiesType"/>
<complexType name="CityObjectGroupPropertiesType">
  <complexContent>
    <extension base="grp:ADEOfCityObjectGroupType">
      <sequence>
        <element minOccurs="0" name="fiscalYearOfPublication" type="gYear"/>
        <element minOccurs="0" name="language" type="gml:CodeType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="CityObjectGroupPropertiesPropertyType">
  <sequence>
    <element ref="uro:CityObjectGroupProperties"/>
  </sequence>
</complexType>
<complexType name="KeyValuePair" substitutionGroup="gml:AbstractObject" type="uro:KeyValuePairType"/>
<complexType name="KeyValuePairType">
  <sequence>
    <element minOccurs="0" name="note" type="string"/>
    <element minOccurs="0" name="surveyYear" type="gYear"/>
  </sequence>
</complexType>
Step 3: CityGML 3.0 building models adopting the converted ADE XML schema

The Project PLATEAU was launched by MLIT (Ministry of Land, Infrastructure, Transport and Tourism) Japan in 2020, and developed 3D city models for more than 10 million buildings of 56 cities, over 10,000km² in Japan. The 3D city models were implemented using the Urban Planning ADE and were published as Open Data (CC BY 4.0/ODbL).

Figure 27 illustrates sample building data, and the Listing 27 shows an example of the GML instance document that was converted from CityGML 2.0 building model to CityGML 3.0. Note that this conversion was operated manually by checking the ADE XML schema carefully and validating the instance document against the XML schema afterwards. Though there are converters for converting CityGML 2.0 data to CityGML 3.0 such as citygml4j, data conversion for ADEs is not yet supported by these converters.
Listing 27. Example of a CityGML 3.0 GML instance document based on the converted ADE XML schema.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel
xmlns:uro="http://www.kantei.go.jp/jp/singi/tiiki/toshisaisei/itoshisaisei/iur/uro/1.4"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.kantei.go.jp/jp/singi/tiiki/toshisaisei/itoshisaisei/iur/uro/1.4 ./urbanObject_CityGML3.xsd">
  <gml:boundedBy>
  </gml:boundedBy>
  <core:cityObjectMember>
    <bldg:Building gml:id="BLD_77ca1a15-3b35-4386-8f86-152ed71c4c64">
      <core:genericAttribute>
      </core:genericAttribute>
      <core:boundary>
      </core:boundary>
      <core:lod0MultiSurface>
      </core:lod0MultiSurface>
      <core:lod1Solid>
      </core:lod1Solid>
      <core:lod2Solid>
      </core:lod2Solid>
      <con:height>
      </con:height>
      <bldg:address>
      </bldg:address>
      <bldg:adeOfAbstractBuilding>
        <uro:BuildingProperties>
          <uro:buildingDetails>
            <uro:BuildingDetails>
              <uro:buildingRoofEdgeArea uom="m2">2513.85520</uro:buildingRoofEdgeArea>
              <uro:districtsAndZonesType codeSpace="../codelists/Common_districtsAndZonesType.xml">4</uro:districtsAndZonesType>
            </uro:BuildingDetails>
            <uro:prefecture>
```
<uro:prefecture codeSpace="../codelists/Common_prefecture.xml">13</uro:prefecture>
<uro:city codeSpace="../codelists/Common_localPublicAuthorities.xml">13104</uro:city>
<uro:surveyYear>2016</uro:surveyYear>
<uro:BuildingDetails>
  <uro:extendedAttribute>
    <uro:KeyValuePair>
      <uro:key codeSpace="../codelists/extendedAttribute_key.xml">2</uro:key>
      <uro:codeValue codeSpace="../codelists/extendedAttribute_key2.xml">2</uro:codeValue>
    </uro:KeyValuePair>
  </uro:extendedAttribute>
  <uro:extendedAttribute>
    <uro:KeyValuePair>
      <uro:key codeSpace="../codelists/extendedAttribute_key.xml">105</uro:key>
      <uro:codeValue codeSpace="../codelists/extendedAttribute_key105.xml">2</uro:codeValue>
    </uro:KeyValuePair>
  </uro:extendedAttribute>
  <uro:extendedAttribute>
    <uro:KeyValuePair>
      <uro:key codeSpace="../codelists/extendedAttribute_key.xml">106</uro:key>
      <uro:codeValue codeSpace="../codelists/extendedAttribute_key106.xml">20</uro:codeValue>
    </uro:KeyValuePair>
  </uro:extendedAttribute>
</uro:BuildingDetails>
</uro:BuildingProperties>
</bldg:adeOfAbstractBuilding>
</bldg:Building>
</core:cityObjectMember>
<core:appearanceMember> ... </core:appearanceMember>
</core:CityModel>
Annex C: XML Schema (Normative)

C.1. Core

The CityGML Core module is defined in the XML Schema Definition file core.xsd (Listing 28). The target namespace http://www.opengis.net/citygml/3.0 is associated with this module.

Listing 28. core.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:core="http://www.opengis.net/citygml/3.0"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xAL="urn:oasis:names:tc:ciq:xal:3"
    elementFormDefault="qualified" targetNamespace="http://www.opengis.net/citygml/3.0"
    version="3.0.0">
  <annotation>
    <documentation>The Core module defines the basic components of the CityGML conceptual model. This includes abstract base classes that define the core properties of more specialized thematic classes defined in other modules as well as concrete classes that are common to other modules, for example basic data types.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/gml/3.2"
    schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <import namespace="urn:oasis:names:tc:ciq:xal:3" schemaLocation="http://docs.oasis-open.org/ciq/v3.0/cs02/xsd/default/xsd/xAL.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net-->  
  <element abstract="true" name="ADEOfAbstractAppearance"
    substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractAppearanceType">
    <annotation>
      <documentation>ADEOfAbstractAppearance acts as a hook to define properties within an ADE that are to be added to AbstractAppearance.</documentation>
    </annotation>
    <element ref="core:ADEOfAbstractAppearance"/>
  </element>
  <complexType abstract="true" name="ADEOfAbstractAppearanceType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractAppearancePropertyType">
    <sequence>
      <element ref="core:ADEOfAbstractAppearance"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfAbstractCityObject"
    substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractCityObjectType">
    <annotation>
      <documentation>ADEOfAbstractCityObject acts as a hook to define properties within an ADE that are to be added to AbstractCityObject.</documentation>
    </annotation>
    <element ref="core:ADEOfAbstractCityObject"/>
  </element>
  <complexType abstract="true" name="ADEOfAbstractCityObjectType">
    <sequence/>
  </complexType>
</schema>
```
<element abstract="true" name="ADEOfAbstractLogicalSpace"
    substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractLogicalSpaceType">
    <annotation>
        <documentation>ADEOfAbstractLogicalSpace acts as a hook to define properties within an ADE that are to be added to AbstractLogicalSpace.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractLogicalSpaceType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractLogicalSpacePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractLogicalSpace"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractOccupiedSpace"
    substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractOccupiedSpaceType">
    <annotation>
        <documentation>ADEOfAbstractOccupiedSpace acts as a hook to define properties within an ADE that are to be added to AbstractOccupiedSpace.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractOccupiedSpaceType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractOccupiedSpacePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractOccupiedSpace"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractPhysicalSpace"
    substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractPhysicalSpaceType">
    <annotation>
        <documentation>ADEOfAbstractPhysicalSpace acts as a hook to define properties within an ADE that are to be added to AbstractPhysicalSpace.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractPhysicalSpaceType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractPhysicalSpacePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractPhysicalSpace"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractPointCloud"
    substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractPointCloudType">
    <annotation>
        <documentation>ADEOfAbstractPointCloud acts as a hook to define properties within an ADE that are to be added to AbstractPointCloud.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractPointCloudType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractPointCloudPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractPointCloud"/>
    </sequence>
</complexType>
within an ADE that are to be added to AbstractPointCloud.

</documentation>
</element>
<complexType abstract="true" name="ADEOfAbstractPointCloudType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractPointCloudPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractPointCloud"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractSpace" substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractSpaceType">
    <annotation>
        <documentation>ADEOfAbstractSpace acts as a hook to define properties within an ADE that are to be added to AbstractSpace.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractSpaceType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractSpacePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractSpace"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractSpaceBoundary" substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractSpaceBoundaryType">
    <annotation>
        <documentation>ADEOfAbstractSpaceBoundary acts as a hook to define properties within an ADE that are to be added to AbstractSpaceBoundary.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractSpaceBoundaryType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractSpaceBoundaryPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractSpaceBoundary"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractThematicSurface" substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractThematicSurfaceType">
    <annotation>
        <documentation>ADEOfAbstractThematicSurface acts as a hook to define properties within an ADE that are to be added to AbstractThematicSurface.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractThematicSurfaceType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractThematicSurfacePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractThematicSurface"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractUnoccupiedSpace"
        substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractUnoccupiedSpaceType">
    <annotation>
        <documentation>ADEOfAbstractUnoccupiedSpace acts as a hook to define properties within an ADE that are to be added to AbstractUnoccupiedSpace.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractUnoccupiedSpaceType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractUnoccupiedSpacePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractUnoccupiedSpace"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractVersion"
        substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractVersionType">
    <annotation>
        <documentation>ADEOfAbstractVersion acts as a hook to define properties within an ADE that are to be added to AbstractVersion.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractVersionType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractVersionPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractVersion"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractVersionTransition"
        substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractVersionTransitionType">
    <annotation>
        <documentation>ADEOfAbstractVersionTransition acts as a hook to define properties within an ADE that are to be added to AbstractVersionTransition.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractVersionTransitionType">
    <sequence/>
</complexType>

<complexType name="ADEOfAbstractVersionTransitionPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractVersionTransition"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAddress" substitutionGroup="gml:AbstractObject" type="core:ADEOfAddressType">
  <annotation>
    <documentation>ADEOfAddress acts as a hook to define properties within an ADE that are to be added to an Address.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfAddressType">
  <sequence/>
</complexType>

<complexType name="ADEOfAddressPropertyType">
  <sequence>
    <element ref="core:ADEOfAddress"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfCityModel" substitutionGroup="gml:AbstractObject" type="core:ADEOfCityModelType">
  <annotation>
    <documentation>ADEOfCityModel acts as a hook to define properties within an ADE that are to be added to a CityModel.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfCityModelType">
  <sequence/>
</complexType>

<complexType name="ADEOfCityModelPropertyType">
  <sequence>
    <element ref="core:ADEOfCityModel"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfClosureSurface" substitutionGroup="gml:AbstractObject" type="core:ADEOfClosureSurfaceType">
  <annotation>
    <documentation>ADEOfClosureSurface acts as a hook to define properties within an ADE that are to be added to a ClosureSurface.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfClosureSurfaceType">
  <sequence/>
</complexType>

<complexType name="ADEOfClosureSurfacePropertyType">
  <sequence>
    <element ref="core:ADEOfClosureSurface"/>
  </sequence>
</complexType>

<element abstract="true" name="AbstractAppearance" substitutionGroup="core:AbstractFeatureWithLifespan" type="core:AbstractAppearanceType">
  <annotation>
    <documentation>AbstractAppearance is the abstract superclass to represent any kind of appearance objects.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractAppearanceType">
  <complexContent>
    <extension base="core:AbstractFeatureWithLifespanType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractAppearance" type="core:ADEOfAbstractAppearancePropertyType">
          <documentation>Augments AbstractAppearance with properties defined in an ADE.</documentation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="AbstractAppearancePropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractAppearance"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element abstract="true" name="AbstractCityObject" substitutionGroup="core:AbstractFeatureWithLifespan" type="core:AbstractCityObjectType">
  <documentation>AbstractCityObject is the abstract superclass of all thematic classes within the CityGML Conceptual Model.</documentation>
</element>

<complexType abstract="true" name="AbstractCityObjectType">
  <complexContent>
    <extension base="core:AbstractFeatureWithLifespanType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="externalReference" type="core:ExternalReferencePropertyType">
          <documentation>References external objects in other information systems that have a relation to the city object.</documentation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="generalizesTo" type="gml:ReferenceType">
          <documentation>Relates generalized representations of the same real-world object in different Levels of Detail to the city object. The direction of this relation is from the city object to the corresponding generalized city objects.</documentation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element minOccurs="0" name="RelativeToTerrain" type="core:RelativeToTerrainType">
    <annotation>
        <documentation>Describes the vertical position of the city object relative to the surrounding terrain.</documentation>
    </annotation>
</element>

<element minOccurs="0" name="RelativeToWater" type="core:RelativeToWaterType">
    <annotation>
        <documentation>Describes the vertical position of the city object relative to the surrounding water surface.</documentation>
    </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="relatedTo">
    <complexType>
        <sequence>
            <element ref="core:CityObjectRelation"/>
        </sequence>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="appearance">
    <annotation>
        <documentation>Relates appearances to the city object.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="core:AbstractAppearance"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="genericAttribute" type="core:AbstractGenericAttributePropertyType">
    <annotation>
        <documentation>Relates generic attributes to the city object.</documentation>
    </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="dynamizer" type="core:AbstractDynamizerPropertyType"/>
Relates Dynamizer objects to the city object. These allow timeseries data to override static attribute values of the city object.

Augments AbstractDynamizer with properties defined in an ADE.

Augments AbstractCityObject with properties defined in an ADE.
AbstractFeature is the abstract superclass of all feature types within the CityGML Conceptual Model.

AbstractFeatureWithLifespan is the base class for all CityGML features. This class allows the optional specification of the real-world and database times for the existence of each feature.

AbstractFeatureWithLifespanWithLifespan is a class that can be used to store additional information about the existence of features.
<annotation>
  <documentation>Indicates the date at which a CityGML feature was removed from the CityModel.</documentation>
</annotation>
</element>
<element minOccurs="0" name="validFrom" type="dateTime">
  <annotation>
    <documentation>Indicates the date at which a CityGML feature started to exist in the real world.</documentation>
  </annotation>
</element>
<element minOccurs="0" name="validTo" type="dateTime">
  <annotation>
    <documentation>Indicates the date at which a CityGML feature ended to exist in the real world.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractFeatureWithLifespan" type="core:ADEOfAbstractFeatureWithLifespanPropertyType">
  <annotation>
    <documentation>Augments AbstractFeatureWithLifespan with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractFeatureWithLifespanPropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractFeatureWithLifespan"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractGenericAttribute" substitutionGroup="gml:AbstractObject" type="core:AbstractGenericAttributeType">
  <annotation>
    <documentation>AbstractGenericAttribute is the abstract superclass for all types of generic attributes.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractGenericAttributeType">
  <sequence/>
</complexType>
<complexType name="AbstractGenericAttributePropertyType">
  <sequence>
    <element ref="core:AbstractGenericAttribute"/>
  </sequence>
</complexType>
<element abstract="true" name="AbstractLogicalSpace" substitutionGroup="core:AbstractSpace" type="core:AbstractLogicalSpaceType">
  <annotation>
    <documentation>AbstractLogicalSpace is the abstract superclass for all types of logical spaces. Logical space refers to spaces that are not bounded by physical surfaces but are defined according to thematic considerations.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractLogicalSpaceType">
  <complexContent>
    <extension base="core:AbstractSpaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractLogicalSpace" type="core:ADEOfAbstractLogicalSpacePropertyType">
          <annotation>
            <documentation>Augments AbstractLogicalSpace with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="AbstractLogicalSpacePropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractLogicalSpace"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element abstract="true" name="AbstractOccupiedSpace" substitutionGroup="core:AbstractPhysicalSpace" type="core:AbstractOccupiedSpaceType">
  <annotation>
    <documentation>AbstractOccupiedSpace is the abstract superclass for all types of physically occupied spaces. Occupied space refers to spaces that are partially or entirely filled with matter.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractOccupiedSpaceType">
  <complexContent>
    <extension base="core:AbstractPhysicalSpaceType">
      <sequence>
        <element minOccurs="0" name="lod1ImplicitRepresentation" type="core:ImplicitGeometryPropertyType">
          <annotation>
            <documentation>Relates to an implicit geometry that represents the occupied space in Level of Detail 1.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<annotation>
  <documentation>Relates to an implicit geometry that represents the
occupied space in Level of Detail 2.</documentation>
</annotation>
</element>
<element minOccurs="0" name="lod3ImplicitRepresentation"
type="core:ImplicitGeometryPropertyType">
  <annotation>
    <documentation>Relates to an implicit geometry that represents the
occupied space in Level of Detail 3.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfAbstractOccupiedSpace" type="core:ADEOfAbstractOccupiedSpacePropertyType">
  <annotation>
    <documentation>Augments AbstractOccupiedSpace with properties defined in
an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractOccupiedSpacePropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractOccupiedSpace"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractPhysicalSpace"
substitutionGroup="core:AbstractSpace" type="core:AbstractPhysicalSpaceType">
  <annotation>
    <documentation>AbstractPhysicalSpace is the abstract superclass for all types of
physical spaces. Physical space refers to spaces that are fully or partially bounded
by physical objects.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractPhysicalSpaceType">
  <complexContent>
    <extension base="core:AbstractSpaceType">
      <sequence>
        <element minOccurs="0" name="lod1TerrainIntersectionCurve"
type="gml:MultiCurvePropertyType">
          <annotation>
            <documentation>Relates to a 3D MultiCurve geometry that represents the
terrain intersection curve of the physical space in Level of Detail 1.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="lod2TerrainIntersectionCurve"
type="gml:MultiCurvePropertyType">
          <annotation>
            <documentation>Relates to a 3D MultiCurve geometry that represents the
terrain intersection curve of the physical space in Level of Detail 2.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<annotation>
<documentation>Relates to a 3D MultiCurve geometry that represents the terrain intersection curve of the physical space in Level of Detail 2.</documentation>
</annotation>
</element>
<element minOccurs="0" name="lod3TerrainIntersectionCurve" type="gml:MultiCurvePropertyType">
<annotation>
<documentation>Relates to a 3D MultiCurve geometry that represents the terrain intersection curve of the physical space in Level of Detail 3.</documentation>
</annotation>
</element>
<element minOccurs="0" name="pointCloud" type="core:AbstractPointCloudPropertyType">
<annotation>
<documentation>Relates to a 3D PointCloud that represents the physical space.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractPhysicalSpace" type="core:ADEOfAbstractPhysicalSpacePropertyType">
<annotation>
<documentation>Augments AbstractPhysicalSpace with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractPhysicalSpacePropertyType">
<complexContent>
<extension base="core:AbstractFeature">
<sequence>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractPointCloud" type="core:ADEOfAbstractPointCloudPropertyType">
<annotation>
<documentation>AbstractPointCloud is the abstract superclass to represent PointCloud objects.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
Augments AbstractPointCloud with properties defined in an ADE.

AbstractSpace is the abstract superclass for all types of spaces. A space is an entity of volumetric extent in the real world.

Specifies the degree of openness of a space.

Specifies qualified volumes related to the space.

Specifies qualified areas related to the space.

Relates to surfaces that bound the space.
<element minOccurs="0" name="lod0Point" type="gml:PointPropertyType">
  <annotation>
    <documentation>Relates to a 3D Point geometry that represents the space in Level of Detail 0.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod0MultiSurface" type="gml:MultiSurfacePropertyType">
  <annotation>
    <documentation>Relates to a 3D MultiSurface geometry that represents the space in Level of Detail 0.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod0MultiCurve" type="gml:MultiCurvePropertyType">
  <annotation>
    <documentation>Relates to a 3D MultiCurve geometry that represents the space in Level of Detail 0.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod1Solid" type="gml:SolidPropertyType">
  <annotation>
    <documentation>Relates to a 3D Solid geometry that represents the space in Level of Detail 1.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod2Solid" type="gml:SolidPropertyType">
  <annotation>
    <documentation>Relates to a 3D Solid geometry that represents the space in Level of Detail 2.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod2MultiSurface" type="gml:MultiSurfacePropertyType">
  <annotation>
    <documentation>Relates to a 3D MultiSurface geometry that represents the space in Level of Detail 2.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod2MultiCurve" type="gml:MultiCurvePropertyType">
  <annotation>
    <documentation>Relates to a 3D MultiCurve geometry that represents the space in Level of Detail 2.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="lod3Solid" type="gml:SolidPropertyType">
  <annotation>
    <documentation>Relates to a 3D Solid geometry that represents the space in Level of Detail 3.</documentation>
  </annotation>
</element>
<complexType name="AbstractSpacePropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractSpace"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element abstract="true" name="AbstractSpaceBoundary" substitutionGroup="core:AbstractCityObject" type="core:AbstractSpaceBoundaryType">
  <annotation>
    <documentation>AbstractSpaceBoundary is the abstract superclass for all types of space boundaries. A space boundary is an entity with areal extent in the real world. Space boundaries are objects that bound a Space. They also realize the contact between adjacent spaces.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractSpaceBoundaryType">
  <complexContent>
    <extension base="core:AbstractCityObjectType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractSpaceBoundary" type="core:ADEOfAbstractSpaceBoundaryPropertyType">
          <annotation>
            <documentation>Augments AbstractSpaceBoundary with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
an ADE.</documentation>
  </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractSpaceBoundaryPropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractSpaceBoundary"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractThematicSurface"
  substitutionGroup="core:AbstractSpaceBoundary"
type="core:AbstractThematicSurfaceType">
  <annotation>
    <documentation>AbstractThematicSurface is the abstract superclass for all types of thematic surfaces.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractThematicSurfaceType">
  <complexContent>
    <extension base="core:AbstractSpaceBoundaryType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="area"
type="core:QualifiedAreaPropertyType">
          <annotation>
            <documentation>Specifies qualified areas related to the thematic surface.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="lod0MultiCurve"
type="gml:MultiCurvePropertyType">
          <annotation>
            <documentation>Relates to a 3D MultiCurve geometry that represents the thematic surface in Level of Detail 0.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="lod0MultiSurface"
type="gml:MultiSurfacePropertyType">
          <annotation>
            <documentation>Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 0.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="lod1MultiSurface"
type="gml:MultiSurfacePropertyType">
          <annotation>
            <documentation>Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 1.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
thematic surface in Level of Detail 1.</documentation>
</element>
<element minOccurs="0" name="lod2MultiSurface" type="gml:MultiSurfacePropertyType">
  <annotation>
    <documentation>Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 2.</documentation>
  </annotation>
</element>
<element minOccurs="0" name="lod3MultiSurface" type="gml:MultiSurfacePropertyType">
  <annotation>
    <documentation>Relates to a 3D MultiSurface geometry that represents the thematic surface in Level of Detail 3.</documentation>
  </annotation>
</element>
<element minOccurs="0" name="pointCloud" type="core:AbstractPointCloudPropertyType">
  <annotation>
    <documentation>Relates to a 3D PointCloud that represents the thematic surface.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractThematicSurface" type="core:ADEOfAbstractThematicSurfacePropertyType">
  <annotation>
    <documentation>Augments AbstractThematicSurface with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractThematicSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractThematicSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<complexType abstract="true" name="AbstractUnoccupiedSpace" substitutionGroup="core:AbstractPhysicalSpace" type="core:AbstractUnoccupiedSpaceType">
  <annotation>
    <documentation>AbstractUnoccupiedSpace is the abstract superclass for all types of physically unoccupied spaces. Unoccupied space refers to spaces that are entirely or mostly free of matter.</documentation>
  </annotation>
</complexType>
<complexType abstract="true" name="AbstractUnoccupiedSpaceType">
  <complexContent>
    <extension base="core:AbstractPhysicalSpaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractUnoccupiedSpace" type="core:ADEOfAbstractUnoccupiedSpacePropertyType">
          <annotation>
            <documentation>Augments AbstractUnoccupiedSpace with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="AbstractUnoccupiedSpacePropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractUnoccupiedSpace"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element abstract="true" name="AbstractVersion" substitutionGroup="core:AbstractFeatureWithLifespan" type="core:AbstractVersionType">
  <annotation>
    <documentation>AbstractVersion is the abstract superclass to represent Version objects.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractVersionType">
  <complexContent>
    <extension base="core:AbstractFeatureWithLifespanType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractVersion" type="core:ADEOfAbstractVersionPropertyType">
          <annotation>
            <documentation>Augments AbstractVersion with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="AbstractVersionPropertyType">
  <sequence minOccurs="0">
    <element ref="core:AbstractVersion"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<complexType abstract="true" name="AbstractVersionTransitionType">
<complexContent>
<extension base="core:AbstractFeatureWithLifespanType">
<sequence>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractVersionTransition" type="core:ADEOfAbstractVersionTransitionPropertyType">
<annotation>
<documentation>Augments AbstractVersionTransition with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AddressType">
<complexContent>
<extension base="core:AbstractFeatureType">
<sequence>
<element name="xalAddress">
<annotation>
<documentation>Relates an OASIS address object to the Address.</documentation>
</annotation>
</element>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexContent>
</complexType>
<element name="Address" substitutionGroup="core:AbstractFeature" type="core:AddressType">
<annotation>
<documentation>Address represents an address of a city object.</documentation>
</annotation>
</element>
<complexType name="AddressType">
<complexContent>
<extension base="core:AbstractFeatureType">
<sequence>
<element name="xalAddress">
<annotation>
<documentation>Relates an OASIS address object to the Address.</documentation>
</annotation>
</element>
</sequence>
</complexType>
</element>
<complexType name="AddressPropertyType">
  <sequence minOccurs="0">
    <element ref="core:Address"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="CityModel" substitutionGroup="core:AbstractFeatureWithLifespan" type="core:CityModelType">
  <annotation>
    <documentation>CityModel is the container for all objects belonging to a city model.</documentation>
  </annotation>
</element>

<complexType name="CityModelType">
  <complexContent>
    <extension base="core:AbstractFeatureWithLifespanType">
      <sequence>
        <element minOccurs="0" name="engineeringCRS">
          <annotation>
            <documentation>Specifies the local engineering coordinate reference system of the CityModel that can be provided inline the CityModel instead of referencing a well-known CRS definition. The definition of an engineering CRS requires an anchor point which relates the origin of the local coordinate system to a point on the earth's surface in order to facilitate the transformation of coordinates from the local engineering CRS.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element minOccurs="0" name="multiPoint" type="gml:MultiPointPropertyType">
  <annotation>
    <documentation>Relates to the MultiPoint geometry of the Address. The geometry relates the address spatially to a city object.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfAddress" type="core:ADEOfAddressPropertyType">
  <annotation>
    <documentation>Augments the Address with properties defined in an ADE.</documentation>
  </annotation>
</element>
<complexType>
  <element name="appearanceMember">
    <annotation>
      <documentation>Specifies the appearances of the CityModel.</documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureMemberType">
          <sequence minOccurs="0">
            <element ref="core:AbstractAppearance"/>
          </sequence>
          <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
      </complexContent>
    </complexType>
  </element>

  <element name="versionMember">
    <annotation>
      <documentation>Specifies the different versions of the CityModel.</documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureMemberType">
          <sequence minOccurs="0">
            <element ref="core:AbstractVersion"/>
          </sequence>
          <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
      </complexContent>
    </complexType>
  </element>

  <element name="versionTransitionMember">
    <annotation>
      <documentation>Specifies the transitions between the different versions of the CityModel.</documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureMemberType">
          <sequence minOccurs="0">
            <element ref="core:AbstractVersionTransition"/>
          </sequence>
          <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
      </complexContent>
    </complexType>
  </element>

  <element name="featureMember">
    <annotation>
      
    </annotation>
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureMemberType">
          <sequence minOccurs="0">
            
          </sequence>
          <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
      </complexContent>
    </complexType>
  </element>
</complexType>
Specifies the feature objects that are part of the CityModel. It allows to include objects that are not derived from a class defined in the CityGML conceptual model, but from the ISO 19109 class AnyFeature.

CityObjectRelation represents a specific relation from the city object in which the relation is included to another city object.

Relates generic attributes to the CityObjectRelation.
ClosureSurface is a special type of thematic surface used to close holes in volumetric objects. Closure surfaces are virtual (non-physical) surfaces. Augments the ClosureSurface with properties defined in an ADE.

DoubleBetween0and1 is a basic type for values, which are greater or equal than 0 and less or equal than 1. The type is used for color encoding, for example.

DoubleBetween0and1List is a basic type that represents a list of double values greater or equal than 0 and less or equal than 1. The type is used for color encoding, for example.
<element name="ExternalReference" substitutionGroup="gml:AbstractObject" type="core:ExternalReferenceType">
  <annotation>
    <documentation>ExternalReference is a reference to a corresponding object in another information system, for example in the German cadastre (ALKIS), the German topographic information system (ATKIS), or the OS UK MasterMap®.</documentation>
  </annotation>
  <complexType name="ExternalReferenceType">
    <sequence>
      <element name="targetResource" type="anyURI">
        <annotation>
          <documentation>Specifies the URI that points to the object in the external information system.</documentation>
        </annotation>
      </element>
      <element minOccurs="0" name="informationSystem" type="anyURI">
        <annotation>
          <documentation>Specifies the URI that points to the external information system.</documentation>
        </annotation>
      </element>
      <element minOccurs="0" name="relationType" type="anyURI">
        <annotation>
          <documentation>Specifies a URI that additionally qualifies the ExternalReference. The URI can point to a definition from an external ontology (e.g. the sameAs relation from OWL) and allows for mapping the ExternalReference to RDF triples.</documentation>
        </annotation>
      </element>
    </sequence>
  </complexType>
  <complexType name="ExternalReferencePropertyType">
    <sequence>
      <element ref="core:ExternalReference"/>
    </sequence>
  </complexType>
</element>

<element name="ImplicitGeometry" substitutionGroup="gml:AbstractGML" type="core:ImplicitGeometryType">
  <annotation>
    <documentation>ImplicitGeometry is a geometry representation where the shape is stored only once as a prototypical geometry. Examples are a tree or other vegetation object, a traffic light or a traffic sign. This prototypic geometry object can be reused or referenced many times, wherever the corresponding feature occurs in the 3D city model.</documentation>
  </annotation>
</element>

<complexType name="ImplicitGeometryType">
  <complexContent>
    164
  </complexContent>
</complexType>
<extension base="gml:AbstractGMLType">
  <sequence>
    <element name="transformationMatrix" type="core:TransformationMatrix4x4Type">
      <annotation>
        <documentation>Specifies the mathematical transformation (translation, rotation, and scaling) between the prototypical geometry and the actual spatial position of the object.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="mimeType" type="gml:CodeType">
      <annotation>
        <documentation>Specifies the MIME type of the external file that stores the prototypical geometry.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="libraryObject" type="anyURI">
      <annotation>
        <documentation>Specifies the URI that points to the prototypical geometry stored in an external file.</documentation>
      </annotation>
    </element>
    <element name="referencePoint" type="gml:PointPropertyType">
      <annotation>
        <documentation>Relates to a 3D Point geometry that represents the base point of the object in the world coordinate system.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="relativeGeometry" type="gml:GeometryPropertyType">
      <annotation>
        <documentation>Relates to a prototypical geometry in a local coordinate system stored inline with the city model.</documentation>
      </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="appearance" type="core:AbstractAppearancePropertyType">
      <annotation>
        <documentation>Relates appearances to the ImplicitGeometry.</documentation>
      </annotation>
    </element>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>
</complexType>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<complexType name="IntegerBetween0and3Type">
<annotation>
<documentation>IntegerBetween0and3 is a basic type for integer values, which are greater or equal than 0 and less or equal than 3. The type is used for encoding the LOD number.</documentation>
</annotation>
<restriction base="integer">
<minInclusive value="0"/>
<maxInclusive value="3"/>
</restriction>
</simpleType>
<element name="Occupy" substitutionGroup="gml:AbstractObject"
type="core:OccupancyType">
<annotation>
<documentation>Occupancy is an application-dependent indication of what is contained by a feature.</documentation>
</annotation>
</element>
<complexType name="OccupancyType">
<sequence>
<element name="numberOfOccupants" type="integer">
<annotation>
<documentation>Indicates the number of occupants contained by a feature.</documentation>
</annotation>
</element>
<element minOccurs="0" name="interval" type="gml:CodeType">
<annotation>
<documentation>Indicates the time period the occupants are contained by a feature.</documentation>
</annotation>
</element>
<element minOccurs="0" name="occupantType" type="gml:CodeType">
<annotation>
<documentation>Indicates the specific type of the occupants that are contained by a feature.</documentation>
</annotation>
</element>
</sequence>
</complexType>
<complexType name="OccupancyPropertyType">
<sequence>
<element ref="core:Occupancy"/>
</sequence>
</complexType>
<element name="QualifiedArea" substitutionGroup="gml:AbstractObject"
type="core:QualifiedAreaType">
<annotation>
<documentation>QualifiedArea is an application-dependent measure of the area of
a space or of a thematic surface.</documentation>
</element>
<complexType name="QualifiedAreaType">
  <sequence>
    <element name="area" type="gml:AreaType">
      <annotation>
        <documentation>Specifies the value of the QualifiedArea.</documentation>
      </annotation>
    </element>
    <element name="typeOfArea" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the specific type of the QualifiedArea.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="QualifiedAreaPropertyType">
  <sequence>
    <element ref="core:QualifiedArea"/>
  </sequence>
</complexType>
<element name="QualifiedVolume" substitutionGroup="gml:AbstractObject" type="core:QualifiedVolumeType">
  <annotation>
    <documentation>QualifiedVolume is an application-dependent measure of the volume of a space.</documentation>
  </annotation>
</element>
<complexType name="QualifiedVolumeType">
  <sequence>
    <element name="volume" type="gml:VolumeType">
      <annotation>
        <documentation>Specifies the value of the QualifiedVolume.</documentation>
      </annotation>
    </element>
    <element name="typeOfVolume" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the specific type of the QualifiedVolume.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="QualifiedVolumePropertyType">
  <sequence>
    <element ref="core:QualifiedVolume"/>
  </sequence>
</complexType>
<complexType name="RelativeToTerrainType"
RelativeToTerrain enumerates the spatial relations of a city object relative to terrain in a qualitative way.

<restriction base="string">
  <enumeration value="entirelyAboveTerrain">
    <annotation>
      <documentation>Indicates that the city object is located entirely above the terrain.</documentation>
    </annotation>
  </enumeration>
  <enumeration value="substantiallyAboveTerrain">
    <annotation>
      <documentation>Indicates that the city object is for the most part located above the terrain.</documentation>
    </annotation>
  </enumeration>
  <enumeration value="substantiallyAboveAndBelowTerrain">
    <annotation>
      <documentation>Indicates that the city object is located half above the terrain and half below the terrain.</documentation>
    </annotation>
  </enumeration>
  <enumeration value="substantiallyBelowTerrain">
    <annotation>
      <documentation>Indicates that the city object is for the most part located below the terrain.</documentation>
    </annotation>
  </enumeration>
  <enumeration value="entirelyBelowTerrain">
    <annotation>
      <documentation>Indicates that the city object is located entirely below the terrain.</documentation>
    </annotation>
  </enumeration>
</restriction>

RelativeToWater enumerates the spatial relations of a city object relative to the water surface in a qualitative way.

<restriction base="string">
  <enumeration value="entirelyAboveWaterSurface">
    <annotation>
      <documentation>Indicates that the city object is located entirely above the water surface.</documentation>
    </annotation>
  </enumeration>
  <enumeration value="substantiallyAboveWaterSurface">
    <annotation>
      <documentation>Indicates that the city object is for the most part located above the water surface.</documentation>
    </annotation>
  </enumeration>
</restriction>
<documentation>Indicates that the city object is for the most part located above the water surface.</documentation>
</annotation>
</enumeration>
<enumeration value="substantiallyAboveAndBelowWaterSurface">
<annotation>
<documentation>Indicates that the city object is located half above the water surface and half below the water surface.</documentation>
</annotation>
</enumeration>
<enumeration value="substantiallyBelowWaterSurface">
<annotation>
<documentation>Indicates that the city object is for the most part located below the water surface.</documentation>
</annotation>
</enumeration>
<enumeration value="entirelyBelowWaterSurface">
<annotation>
<documentation>Indicates that the city object is located entirely below the water surface.</documentation>
</annotation>
</enumeration>
<enumeration value="temporarilyAboveAndBelowWaterSurface">
<annotation>
<documentation>Indicates that the city object is temporarily located above or below the water level, because the height of the water surface is varying.</documentation>
</annotation>
</enumeration>
</restriction>
</simpleType>
<simpleType name="SpaceTypeType">
<annotation>
<documentation>SpaceType is an enumeration that characterises a space according to its closure properties.</documentation>
</annotation>
<restriction base="string">
<enumeration value="closed">
<annotation>
<documentation>Indicates that the space has boundaries at the bottom, at the top, and on all sides.</documentation>
</annotation>
</enumeration>
<enumeration value="open">
<annotation>
<documentation>Indicates that the space has at maximum a boundary at the bottom.</documentation>
</annotation>
</enumeration>
<enumeration value="semiOpen">
<annotation>
<documentation>Indicates that the space has boundaries at the bottom, but not at the top or on all sides.</documentation>
</annotation>
</enumeration>
</restriction>
</simpleType>
C.2. Appearance

The CityGML Appearance module is defined in the XML Schema Definition file appearance.xsd (Listing 29). The target namespace http://www.opengis.net/citygml/appearance/3.0 is associated with this module.

Listing 29. appearance.xsd

```xml
  <annotation>
    <documentation>The Appearance module supports the modelling of the observable...
  </annotation>
</schema>
```
surface properties of CityGML features in the form of textures and material. 

```xml
<element abstract="true" name="ADEOfAbstractSurfaceData" substitutionGroup="gml:AbstractObject" type="app:ADEOfAbstractSurfaceDataType">
  <documentation>ADEOfAbstractSurfaceData acts as a hook to define properties within an ADE that are to be added to AbstractSurfaceData.</documentation>
</element>

<complexType abstract="true" name="ADEOfAbstractSurfaceDataType">
  <sequence/>
</complexType>

<complexType name="ADEOfAbstractSurfaceDataPropertyType">
  <sequence>
    <element ref="app:ADEOfAbstractSurfaceData"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractTexture" substitutionGroup="gml:AbstractObject" type="app:ADEOfAbstractTextureType">
  <documentation>ADEOfAbstractTexture acts as a hook to define properties within an ADE that are to be added to AbstractTexture.</documentation>
</element>

<complexType abstract="true" name="ADEOfAbstractTextureType">
  <sequence/>
</complexType>

<complexType name="ADEOfAbstractTexturePropertyType">
  <sequence>
    <element ref="app:ADEOfAbstractTexture"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfAppearance" substitutionGroup="gml:AbstractObject" type="app:ADEOfAppearanceType">
  <documentation>ADEOfAppearance acts as a hook to define properties within an ADE that are to be added to an Appearance.</documentation>
</element>

<complexType abstract="true" name="ADEOfAppearanceType">
  <sequence/>
</complexType>

<complexType name="ADEOfAppearancePropertyType">
  <sequence>
    <element ref="app:ADEOfAppearance"/>
  </sequence>
</complexType>
```
<complexType abstract="true" name="ADEOfGeoreferencedTexture"
substitutionGroup="gml:AbstractObject" type="app:ADEOfGeoreferencedTextureType">  
  <annotation>  
    <documentation>ADEOfGeoreferencedTexture acts as a hook to define properties within an ADE that are to be added to a GeoreferencedTexture.</documentation>  
  </annotation>  
</element>  
</complexType>  
<complexType abstract="true" name="ADEOfGeoreferencedTextureType">  
  <sequence/>  
</complexType>  
<complexType name="ADEOfGeoreferencedTexturePropertyType">  
  <sequence>  
    <element ref="app:ADEOfGeoreferencedTexture"/>  
  </sequence>  
</complexType>  
<element abstract="true" name="ADEOfParameterizedTexture"
substitutionGroup="gml:AbstractObject" type="app:ADEOfParameterizedTextureType">  
  <annotation>  
    <documentation>ADEOfParameterizedTexture acts as a hook to define properties within an ADE that are to be added to a ParameterizedTexture.</documentation>  
  </annotation>  
</element>  
</complexType>  
<complexType abstract="true" name="ADEOfParameterizedTextureType">  
  <sequence/>  
</complexType>  
<complexType name="ADEOfParameterizedTexturePropertyType">  
  <sequence>  
    <element ref="app:ADEOfParameterizedTexture"/>  
  </sequence>  
</complexType>  
<element abstract="true" name="ADEOfX3DMaterial"
substitutionGroup="gml:AbstractObject" type="app:ADEOfX3DMaterialType">  
  <annotation>  
    <documentation>ADEOfX3DMaterial acts as a hook to define properties within an ADE that are to be added to an X3DMaterial.</documentation>  
  </annotation>  
</element>  
</complexType>  
<complexType abstract="true" name="ADEOfX3DMaterialType">  
  <sequence/>  
</complexType>  
<complexType name="ADEOfX3DMaterialPropertyType">  
  <sequence>  
    <element ref="app:ADEOfX3DMaterial"/>  
  </sequence>  
</complexType>  
<element abstract="true" name="AbstractSurfaceData"
substitutionGroup="core:AbstractFeature" type="app:AbstractSurfaceDataType">  
  <annotation>  
    <documentation>AbstractSurfaceData is the abstract superclass for different kinds of textures and material.</documentation>  
  </annotation>  
</element>
complexType abstract="true" name="AbstractSurfaceDataType">
      <complexContent>
        <extension base="core:AbstractFeatureType">
          <sequence>
            <element default="true" minOccurs="0" name="isFront" type="boolean">
              <annotation>
                <documentation>Indicates whether the texture or material is assigned to the front side or the back side of the surface geometry object.</documentation>
              </annotation>
            </element>
            <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractSurfaceData" type="app:ADEOfAbstractSurfaceDataPropertyType">
              <annotation>
                <documentation>Augments AbstractSurfaceData with properties defined in an ADE.</documentation>
              </annotation>
            </element>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </complexType>
  <complexType name="AbstractSurfaceDataPropertyType">
    <sequence minOccurs="0">
      <element ref="app:AbstractSurfaceData"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </complexType>
  <element abstract="true" name="AbstractTexture" substitutionGroup="app:AbstractSurfaceData" type="app:AbstractTextureType">
    <annotation>
      <documentation>AbstractTexture is the abstract superclass to represent the common attributes of the classes ParameterizedTexture and GeoreferencedTexture.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="AbstractTextureType">
    <complexContent>
      <extension base="app:AbstractSurfaceDataType">
        <sequence>
          <element name="imageURI" type="anyURI">
            <annotation>
              <documentation>Specifies the URI that points to the external image data file.</documentation>
            </annotation>
          </element>
          <element minOccurs="0" name="mimeType" type="gml:CodeType">
            <annotation>
              <documentation>Specifies the MIME type of the external point cloud file.</documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<element minOccurs="0" name="textureType" type="app:TextureTypeType">
  <annotation>
    <documentation>Indicates the specific type of the texture.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="wrapMode" type="app:WrapModeType">
  <annotation>
    <documentation>Specifies the behaviour of the texture when the texture is smaller than the surface to which it is applied.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="borderColor" type="app:ColorPlusOpacityType">
  <annotation>
    <documentation>Specifies the color of that part of the surface that is not covered by the texture.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractTexture" type="app:ADEOfAbstractTexturePropertyType">
  <annotation>
    <documentation>Augments AbstractTexture with properties defined in an ADE.</documentation>
  </annotation>
</element>

<complexType name="AbstractTexturePropertyType">
  <sequence minOccurs="0">
    <element ref="app:AbstractTexture"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<complexType abstract="true" name="AbstractTextureParameterization" substitutionGroup="gml:AbstractObject" type="app:AbstractTextureParameterizationType">
  <annotation>
    <documentation>AbstractTextureParameterization is the abstract superclass for different kinds of texture parameterizations.</documentation>
  </annotation>
</complexType>

<complexType abstract="true" name="AbstractTextureParameterizationType">
  <sequence/>
</complexType>

<complexType name="AbstractTextureParameterizationPropertyType">
  <sequence>
  </sequence>
</complexType>
<element ref="app:AbstractTextureParameterization"/>
</sequence>
</complexType>
<element name="Appearance" substitutionGroup="core:AbstractAppearance" type="app:AppearanceType">
    <annotation>
        <documentation>An Appearance is a collection of surface data, i.e. observable properties for surface geometry objects in the form of textures and material.</documentation>
    </annotation>
</element>
<complexType name="AppearanceType">
    <complexContent>
        <extension base="core:AbstractAppearanceType">
            <sequence>
                <element minOccurs="0" name="theme" type="string">
                    <annotation>
                        <documentation>Specifies the topic of the Appearance. Each Appearance contains surface data for one theme only. Examples of themes are infrared radiation, noise pollution, or earthquake-induced structural stress.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="surfaceData">
                    <annotation>
                        <documentation>Relates to the surface data that are part of the Appearance.</documentation>
                    </annotation>
                </element>
                <element minOccurs="0" name="adeOfAppearance" type="app:ADEOfAppearancePropertyType">
                    <annotation>
                        <documentation>Augments the Appearance with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element ref="app:Appearance"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<complexType name="ColorType">
<annotation>
<documentation>Color is a list of three double values between 0 and 1 defining an RGB color value.</documentation>
</annotation>
<restriction base="core:DoubleBetween0and1ListType">
<length value="3"/>
</restriction>
</complexType>
<complexType name="ColorPlusOpacityType">
<annotation>
<documentation>Color is a list of four double values between 0 and 1 defining an RGBA color value. Opacity value of 0 means transparent.</documentation>
</annotation>
<restriction base="core:DoubleBetween0and1ListType">
<minLength value="3"/>
<maxLength value="4"/>
</restriction>
</complexType>
<element name="GeoreferencedTexture" substitutionGroup="app:AbstractTexture" type="app:GeoreferencedTextureType">
<annotation>
<documentation>A GeoreferencedTexture is a texture that uses a planimetric projection. It contains an implicit parameterization that is either stored within the image file, an accompanying world file or specified using the orientation and referencePoint elements.</documentation>
</annotation>
</element>
<complexType name="GeoreferencedTextureType">
<complexContent>
<extension base="app:AbstractTextureType">
<sequence>
<element default="true" minOccurs="0" name="preferWorldFile" type="boolean">
<annotation>
<documentation>Indicates whether the georeference from the image file or the accompanying world file should be preferred.</documentation>
</annotation>
</element>
<element minOccurs="0" name="referencePoint" type="gml:PointPropertyType">
<annotation>
<documentation>Relates to the 2D Point geometry that represents the center of the upper left image pixel in world space.</documentation>
</annotation>
</element>
<element minOccurs="0" name="orientation" type="core:TransformationMatrix2x2Type">
<documentation>Specifies the rotation and scaling of the image in form of a 2x2 matrix.</documentation>

<element maxOccurs="unbounded" minOccurs="0" name="target" type="anyURI">
  <documentation>Specifies the URI that points to the surface geometry objects to which the texture is applied.</documentation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfGeoreferencedTexture" type="app:ADEOfGeoreferencedTexturePropertyType">
  <documentation>Augments the GeoreferencedTexture with properties defined in an ADE.</documentation>
</element>

<complexType name="ParameterizedTextureType">
  <complexContent>
    <extension base="app:AbstractTextureType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="textureParameterization">
          <documentation>Relates to the texture coordinates or transformation matrices used for parameterization.</documentation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0"
  name="adeOfParameterizedTexture" type="app:ADEOfParameterizedTexturePropertyType">
  <annotation>
    <documentation>Augments the ParameterizedTexture with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="ParameterizedTexturePropertyType">
  <sequence minOccurs="0">
    <element ref="app:ParameterizedTexture"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TexCoordGen" substitutionGroup="app:AbstractTextureParameterization" type="app:TexCoordGenType">
  <annotation>
    <documentation>TexCoordGen defines texture parameterization using a transformation matrix.</documentation>
  </annotation>
</element>
<complexType name="TexCoordGenType">
  <complexContent>
    <extension base="app:AbstractTextureParameterizationType">
      <sequence>
        <element name="worldToTexture" type="core:TransformationMatrix3x4Type">
          <annotation>
            <documentation>Specifies the 3x4 transformation matrix that defines the transformation between world coordinates and texture coordinates.</documentation>
          </annotation>
        </element>
      </sequence>
      <attributeGroup ref="gml:SRSReferenceGroup"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="TexCoordGenPropertyType">
  <sequence>
    <element ref="app:TexCoordGen"/>
  </sequence>
</complexType>
<element name="TexCoordList" substitutionGroup="app:AbstractTextureParameterization" type="app:TexCoordListType">
  <annotation>
    <documentation>TexCoordList defines texture parameterization using texture coordinates.</documentation>
  </annotation>
</element>
<complexType name="TexCoordListType">
    <complexContent>
        <extension base="app:AbstractTextureParameterizationType">
            <sequence>
                <element maxOccurs="unbounded" name="textureCoordinates" type="gml:doubleList">
                    <annotation>
                        <documentation>
                            Specifies the coordinates of texture used for parameterization. The texture coordinates are provided separately for each LinearRing of the surface geometry object.
                        </documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" name="ring" type="anyURI">
                    <annotation>
                        <documentation>
                            Specifies the URIs that point to the LinearRings that are parameterized using the given texture coordinates.
                        </documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="TexCoordListPropertyType">
    <sequence>
        <element ref="app:TexCoordList"/>
    </sequence>
</complexType>

<element name="TextureAssociation" substitutionGroup="gml:AbstractGML" type="app:TextureAssociationType">
    <annotation>
        <documentation>
            TextureAssociation denotes the relation of a texture to a surface geometry object.
        </documentation>
    </annotation>
</element>

<complexType name="TextureAssociationType">
    <complexContent>
        <extension base="gml:AbstractGMLType">
            <sequence>
                <element name="target" type="anyURI">
                    <annotation>
                        <documentation>
                            Specifies the URI that points to the surface geometry object to which the texture is applied.
                        </documentation>
                    </annotation>
                </element>
                <element name="textureParameterization" type="app:AbstractTextureParameterizationPropertyType">
                    <annotation>
                        <documentation>
                            Relates to the texture coordinates or transformation matrices used for parameterization.
                        </documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="TextureAssociationPropertyType">
  <sequence minOccurs="0">
    <element ref="app:TextureAssociation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<complexType name="TextureTypeType">
  <annotation>
    <documentation>TextureType enumerates the different texture types.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="specific">
      <annotation>
        <documentation>Indicates that the texture is specific to a single surface.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="typical">
      <annotation>
        <documentation>Indicates that the texture is characteristic of a surface and can be used repeatedly.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="unknown">
      <annotation>
        <documentation>Indicates that the texture type is not known.</documentation>
      </annotation>
    </enumeration>
  </restriction>
</complexType>

<complexType name="WrapModeType">
  <annotation>
    <documentation>WrapMode enumerates the different fill modes for textures.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="none">
      <annotation>
        <documentation>Indicates that the texture is applied to the surface "as is". The part of the surface that is not covered by the texture is shown fully transparent. [cf. COLLADA]</documentation>
      </annotation>
    </enumeration>
    <enumeration value="wrap">
      ...
    </enumeration>
  </restriction>
</complexType>
<annotation>
  <documentation>Indicates that the texture is repeated until the surface is fully covered. [cf. COLLADA]</documentation>
</annotation>
</enumeration>
<enumeration value="mirror">
  <annotation>
    <documentation>Indicates that the texture is repeated and mirrored. [cf. COLLADA]</documentation>
  </annotation>
</enumeration>
<enumeration value="clamp">
  <annotation>
    <documentation>Indicates that the texture is stretched to the edges of the surface. [cf. COLLADA]</documentation>
  </annotation>
</enumeration>
<enumeration value="border">
  <annotation>
    <documentation>Indicates that the texture is applied to the surface "as is". The part of the surface that is not covered by the texture is filled with the RGBA color that is specified in the attribute borderColor. [cf. COLLADA]</documentation>
  </annotation>
</enumeration>
</restriction>
</simpleType>
<element name="X3DMaterial" substitutionGroup="app:AbstractSurfaceData" type="app:X3DMaterialType">
  <annotation>
    <documentation>X3DMaterial defines properties for surface geometry objects based on the material definitions from the X3D and COLLADA standards.</documentation>
  </annotation>
</element>
<complexType name="X3DMaterialType">
  <complexContent>
    <extension base="app:AbstractSurfaceDataType">
      <sequence>
        <element default="0.2" minOccurs="0" name="ambientIntensity" type="core:DoubleBetween0and1Type">
          <annotation>
            <documentation>Specifies the minimum percentage of diffuseColor that is visible regardless of light sources.</documentation>
          </annotation>
        </element>
        <element default="0.8 0.8 0.8" minOccurs="0" name="diffuseColor" type="app:ColorType">
          <annotation>
            <documentation>Specifies the color of the light diffusely reflected by the surface geometry object.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element default="0.0 0.0 0.0" minOccurs="0" name="emissiveColor" type="app:ColorType">
  <annotation>
    <documentation>Specifies the color of the light emitted by the surface geometry object.</documentation>
  </annotation>
</element>

<element default="1.0 1.0 1.0" minOccurs="0" name="specularColor" type="app:ColorType">
  <annotation>
    <documentation>Specifies the color of the light directly reflected by the surface geometry object.</documentation>
  </annotation>
</element>

<element default="0.2" minOccurs="0" name="shininess" type="core:DoubleBetween0and1Type">
  <annotation>
    <documentation>Specifies the sharpness of the specular highlight.</documentation>
  </annotation>
</element>

<element default="0.0" minOccurs="0" name="transparency" type="core:DoubleBetween0and1Type">
  <annotation>
    <documentation>Specifies the degree of transparency of the surface geometry object.</documentation>
  </annotation>
</element>

<element default="false" minOccurs="0" name="isSmooth" type="boolean">
  <annotation>
    <documentation>Specifies which interpolation method is used for the shading of the surface geometry object. If the attribute is set to true, vertex normals should be used for shading (Gouraud shading). Otherwise, normals should be constant for a surface patch (flat shading).</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="target" type="anyURI">
  <annotation>
    <documentation>Specifies the URI that points to the surface geometry objects to which the material is applied.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfX3DMaterial" type="app:ADEOfX3DMaterialPropertyType">
  <annotation>
    <documentation>Augments the X3DMaterial with properties defined in an ADE.</documentation>
  </annotation>
</element>
C.3. Bridge

The CityGML Bridge module is defined in the XML Schema Definition file bridge.xsd (Listing 30). The target namespace http://www.opengis.net/citygml/bridge/3.0 is associated with this module.

Listing 30. bridge.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:brid="http://www.opengis.net/citygml/bridge/3.0"
   xmlns:con="http://www.opengis.net/citygml/construction/3.0"
   xmlns:core="http://www.opengis.net/citygml/3.0"
   xmlns:gml="http://www.opengis.net/gml/3.2"
   elementFormDefault="qualified"
   targetNamespace="http://www.opengis.net/citygml/bridge/3.0"
   version="3.0.0">
  <annotation>
    <documentation>The Bridge module supports representation of thematic and spatial aspects of bridges, bridge parts, bridge installations, and interior bridge structures.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
  <import namespace="http://www.opengis.net/citygml/construction/3.0" schemaLocation="./construction.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net-->
  <element abstract="true" name="ADEOfAbstractBridge" substitutionGroup="gml:AbstractObject" type="brid:ADEOfAbstractBridgeType">
    <annotation>
      <documentation>ADEOfAbstractBridge acts as a hook to define properties within an ADE that are to be added to AbstractBridge.</documentation>
    </annotation>
    <element>
      <complexType abstract="true" name="ADEOfAbstractBridgeType">
        <sequence/>
      </complexType>
      <complexType name="ADEOfAbstractBridgePropertyType">
        <sequence>
          <element ref="brid:ADEOfAbstractBridge"/>
        </sequence>
      </complexType>
    </element>
  </element>
</schema>
```
<element abstract="true" name="ADEOfBridge" substitutionGroup="gml:AbstractObject" type="brid:ADEOfBridgeType">
  <annotation>
    <documentation>ADEOfBridge acts as a hook to define properties within an ADE that are to be added to a Bridge.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBridgeType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgePropertyType">
  <sequence>
    <element ref="brid:ADEOfBridge"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBridgeConstructiveElement" substitutionGroup="gml:AbstractObject" type="brid:ADEOfBridgeConstructiveElementType">
  <annotation>
    <documentation>ADEOfBridgeConstructiveElement acts as a hook to define properties within an ADE that are to be added to a BridgeConstructiveElement.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBridgeConstructiveElementType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgeConstructiveElementPropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgeConstructiveElement"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBridgeFurniture" substitutionGroup="gml:AbstractObject" type="brid:ADEOfBridgeFurnitureType">
  <annotation>
    <documentation>ADEOfBridgeFurniture acts as a hook to define properties within an ADE that are to be added to a BridgeFurniture.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBridgeFurnitureType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgeFurniturePropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgeFurniture"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBridgeInstallation" substitutionGroup="gml:AbstractObject" type="brid:ADEOfBridgeInstallationType">
  <annotation>
    <documentation>ADEOfBridgeInstallation acts as a hook to define properties within an ADE that are to be added to a BridgeInstallation.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBridgeInstallationType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgeInstallationPropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgeInstallation"/>
  </sequence>
</complexType>

<complexType abstract="true" name="ADEOfBridgeInstallationType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgeInstallationPropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgeInstallation"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBridgePart" substitutionGroup="gml:AbstractObject" type="brid:ADEOfBridgePartType">
  <annotation>
    <documentation>ADEOfBridgePart acts as a hook to define properties within an ADE that are to be added to a BridgePart.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBridgePartType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgePartPropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgePart"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBridgeRoom" substitutionGroup="gml:AbstractObject" type="brid:ADEOfBridgeRoomType">
  <annotation>
    <documentation>ADEOfBridgeRoom acts as a hook to define properties within an ADE that are to be added to a BridgeRoom.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBridgeRoomType">
  <sequence/>
</complexType>
<complexType name="ADEOfBridgeRoomPropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgeRoom"/>
  </sequence>
</complexType>
<element abstract="true" name="AbstractBridge" substitutionGroup="con:AbstractConstruction" type="brid:AbstractBridgeType">
  <annotation>
    <documentation>AbstractBridge is an abstract superclass representing the common attributes and associations of the classes Bridge and BridgePart.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractBridgeType">
  <complexContent>
    <extension base="con:AbstractConstructionType">
      <sequence>
        ...
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element minOccurs="0" name="class" type="gml:CodeType">
  <annotation>
    <documentation>Indicates the specific type of the Bridge or BridgePart.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the intended purposes of the Bridge or BridgePart.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the actual uses of the Bridge or BridgePart.</documentation>
  </annotation>
</element>

<element default="false" minOccurs="0" name="isMovable" type="boolean">
  <annotation>
    <documentation>Indicates whether the Bridge or BridgePart can be moved to allow for watercraft to pass.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="bridgeConstructiveElement">
  <annotation>
    <documentation>Relates the constructive elements to the Bridge or BridgePart.</documentation>
  </annotation>
</element>

<complexType>
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="brid:BridgeConstructiveElement"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element maxOccurs="unbounded" minOccurs="0" name="bridgeInstallation">
  <annotation>
    <documentation>Relates the installation objects to the Bridge or BridgePart.</documentation>
  </annotation>
</element>

<complexType>
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="brid:BridgeConstructiveElement"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<sequence minOccurs="0">
  <element ref="brid:BridgeInstallation"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="bridgeRoom">
  <annotation>
    <documentation>Relates the rooms to the Bridge or BridgePart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="brid:BridgeRoom"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="bridgeFurniture">
  <annotation>
    <documentation>Relates the furniture objects to the Bridge or BridgePart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="brid:BridgeFurniture"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="address" type="core:AddressPropertyType">
  <annotation>
    <documentation>Relates the addresses to the Bridge or BridgePart.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractBridge" type="brid:ADEOfAbstractBridgePropertyType">
  <annotation>
    <documentation>Augments AbstractBridge with properties defined in an
A Bridge represents a structure that affords the passage of pedestrians, animals, vehicles, and service(s) above obstacles or between two points at a height above ground. [cf. ISO 6707-1]
<complexType name="BridgePropertyType">
  <sequence minOccurs="0">
    <element ref="brid:Bridge"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BridgeConstructiveElement" substitutionGroup="con:AbstractConstructiveElement" type="brid:BridgeConstructiveElementType">
    <annotation>
      <documentation>A BridgeConstructiveElement is an element of a bridge which is essential from a structural point of view. Examples are pylons, anchorages, slabs, beams.</documentation>
    </annotation>
</element>

<complexType name="BridgeConstructiveElementType">
  <complexContent>
    <extension base="con:AbstractConstructiveElementType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the BridgeConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the BridgeConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the BridgeConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfBridgeConstructiveElement" type="brid:ADEOfBridgeConstructiveElementPropertyType">
          <annotation>
            <documentation>Augments the BridgeConstructiveElement with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BridgeConstructiveElementPropertyType">
  <sequence minOccurs="0">
    <element ref="brid:BridgeConstructiveElement"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BridgeFurniture" substitutionGroup="con:AbstractFurniture" type="brid:BridgeFurnitureType">
  <annotation>
    <documentation>A BridgeFurniture is an equipment for occupant use, usually not fixed to the bridge. [cf. ISO 6707-1]</documentation>
  </annotation>
</element>

<complexType name="BridgeFurnitureType">
  <complexContent>
    <extension base="con:AbstractFurnitureType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the BridgeFurniture.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the BridgeFurniture.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the BridgeFurniture.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfBridgeFurniture" type="brid:ADEOfBridgeFurniturePropertyType">
          <annotation>
            <documentation>Augments the BridgeFurniture with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BridgeFurniturePropertyType">
  <sequence minOccurs="0">
    <element ref="brid:BridgeFurniture"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BridgeInstallation" substitutionGroup="con:AbstractInstallation" type="brid:BridgeInstallationType">
  <annotation>
    <documentation>A BridgeInstallation is a permanent part of a Bridge (inside and/or outside) which does not have the significance of a BridgePart. In contrast to BridgeConstructiveElements, a BridgeInstallation is not essential from a structural point of view. Examples are stairs, antennas or railways.</documentation>
  </annotation>
</element>

<complexType name="BridgeInstallationType">
  <complexContent>
    <extension base="con:AbstractInstallationType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the BridgeInstallation.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the BridgeInstallation.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the BridgeInstallation.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfBridgeInstallation" type="brid:ADEOfBridgeInstallationPropertyType">
          <annotation>
            <documentation>Augments the BridgeInstallation with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BridgeInstallationPropertyType">
    <sequence minOccurs="0">
        <element ref="brid:BridgeInstallation"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BridgePart" substitutionGroup="brid:AbstractBridge"
type="brid:BridgePartType">
    <annotation>
        <documentation>A BridgePart is a physical or functional subdivision of a Bridge. It would be considered a Bridge, if it were not part of a collection of other BridgeParts.</documentation>
    </annotation>
</element>

<complexType name="BridgePartType">
    <complexContent>
        <extension base="brid:AbstractBridgeType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfBridgePart" type="brid:ADEOfBridgePartPropertyType">
                    <annotation>
                        <documentation>Augments the BridgePart with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="BridgePartPropertyType">
    <sequence minOccurs="0">
        <element ref="brid:BridgePart"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BridgeRoom" substitutionGroup="core:AbstractUnoccupiedSpace"
type="brid:BridgeRoomType">
    <annotation>
        <documentation>A BridgeRoom is a space within a Bridge or BridgePart intended for human occupancy (e.g. a place of work or recreation) and/or containment (storage) of animals or things. A BridgeRoom is bounded physically and/or virtually (e.g. by ClosureSurfaces or GenericSurfaces).</documentation>
    </annotation>
</element>

<complexType name="BridgeRoomType">
    <complexContent>
        <extension base="core:AbstractUnoccupiedSpaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    192
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<annotation>
  <documentation>Indicates the specific type of the BridgeRoom.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the intended purposes of the BridgeRoom.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the actual uses of the BridgeRoom.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="bridgeFurniture">
  <annotation>
    <documentation>Relates the furniture objects to the BridgeRoom.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="brid:BridgeFurniture"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="bridgeInstallation">
  <annotation>
    <documentation>Relates to the installation objects to the BridgeRoom.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="brid:BridgeInstallation"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>
C.4. Building

The CityGML Building module is defined in the XML Schema Definition file building.xsd (Listing 31). The target namespace http://www.opengis.net/citygml/building/3.0 is associated with this module.

Listing 31. building.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:bldg="http://www.opengis.net/citygml/building/3.0"
   xmlns:con="http://www.opengis.net/citygml/construction/3.0"
   xmlns:core="http://www.opengis.net/citygml/3.0"
   xmlns:gml="http://www.opengis.net/gml/3.2"
   elementFormDefault="qualified"
   targetNamespace="http://www.opengis.net/citygml/building/3.0" version="3.0.0">
   <annotation>
     <documentation>The Building module supports representation of thematic and spatial aspects of buildings, building parts, building installations, building subdivisions, and interior building structures.</documentation>
   </annotation>
   <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
   <import namespace="http://www.opengis.net/citygml/construction/3.0" schemaLocation="./construction.xsd"/>
   <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
   <!--XML Schema document created by ShapeChange - http://shapechange.net/-->
   <element abstract="true" name="ADEOfAbstractBuilding"
      substitutionGroup="gml:AbstractObject" type="bldg:ADEOfAbstractBuildingType">
     <annotation>
      <documentation>ADEOfAbstractBuilding acts as a hook to define properties within an ADE that are to be added to AbstractBuilding.</documentation>
     </annotation>
   </element>
</schema>
```
<complexType abstract="true" name="ADEOfAbstractBuildingType">
  <sequence/>
</complexType>

<complexType name="ADEOfAbstractBuildingPropertyType">
  <sequence>
    <element ref="bldg:ADEOfAbstractBuilding"/>
  </sequence>
</complexType>

<complexType abstract="true" name="ADEOfAbstractBuildingSubdivision" substitutionGroup="gml:AbstractObject" type="bldg:ADEOfAbstractBuildingSubdivisionType">
  <annotation>
    <documentation>ADEOfAbstractBuildingSubdivision acts as a hook to define properties within an ADE that are to be added to AbstractBuildingSubdivision.</documentation>
  </annotation>
</complexType>

<complexType abstract="true" name="ADEOfAbstractBuildingSubdivisionPropertyType">
  <sequence>
    <element ref="bldg:ADEOfAbstractBuildingSubdivision"/>
  </sequence>
</complexType>

<complexType name="ADEOfBuilding" substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingType">
  <annotation>
    <documentation>ADEOfBuilding acts as a hook to define properties within an ADE that are to be added to a Building.</documentation>
  </annotation>
</complexType>

<complexType abstract="true" name="ADEOfBuildingPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuilding"/>
  </sequence>
</complexType>

<complexType abstract="true" name="ADEOfBuildingConstructiveElement" substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingConstructiveElementType">
  <annotation>
    <documentation>ADEOfBuildingConstructiveElement acts as a hook to define properties within an ADE that are to be added to a BuildingConstructiveElement.</documentation>
  </annotation>
</complexType>
<complexType name="ADEOfBuildingConstructiveElementPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingConstructiveElement"/>
  </sequence>
</complexType>

<complexType name="ADEOfBuildingFurniturePropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingFurniture"/>
  </sequence>
</complexType>

<complexType name="ADEOfBuildingInstallationPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingInstallation"/>
  </sequence>
</complexType>

<complexType name="ADEOfBuildingPartPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingPart"/>
  </sequence>
</complexType>
<complexType name="ADEOfBuildingRoomPropertyType">
<sequence>
  <element ref="bldg:ADEOfBuildingRoom"/>
</sequence>
</complexType>

<element abstract="true" name="ADEOfBuildingUnit" substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingUnitType">
<annotation>
  <documentation>ADEOfBuildingUnit acts as a hook to define properties within an ADE that are to be added to a BuildingUnit.</documentation>
</annotation>
</element>
<complexType abstract="true" name="ADEOfBuildingUnitType">
<sequence/>
</complexType>
<complexType name="ADEOfBuildingUnitPropertyType">
<sequence>
  <element ref="bldg:ADEOfBuildingUnit"/>
</sequence>
</complexType>

<element abstract="true" name="ADEOfStorey" substitutionGroup="gml:AbstractObject" type="bldg:ADEOfStoreyType">
<annotation>
  <documentation>ADEOfStorey acts as a hook to define properties within an ADE that are to be added to a Storey.</documentation>
</annotation>
</element>
<complexType abstract="true" name="ADEOfStoreyType">
<sequence/>
</complexType>
<complexType name="ADEOfStoreyPropertyType">
<sequence>
  <element ref="bldg:ADEOfStorey"/>
</sequence>
</complexType>

<element abstract="true" name="AbstractBuilding" substitutionGroup="con:AbstractConstruction" type="bldg:AbstractBuildingType">
<annotation>
  <documentation>AbstractBuilding is an abstract superclass representing the common attributes and associations of the classes Building and
</documentation>
</annotation>
</element>
<complexType abstract="true" name="AbstractBuildingType">
  <complexContent>
    <extension base="con:AbstractConstructionType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the Building or BuildingPart.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the Building or BuildingPart.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the Building or BuildingPart.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="roofType" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the shape of the roof of the Building or BuildingPart.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="storeysAboveGround" type="integer">
          <annotation>
            <documentation>Indicates the number of storeys positioned above ground level.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="storeysBelowGround" type="integer">
          <annotation>
            <documentation>Indicates the number of storeys positioned below ground level.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="storeyHeightsAboveGround" type="gml:MeasureOrNilReasonListType">
          <annotation>
            <documentation>Lists the heights of each storey above ground. The first value in the list denotes the height of the storey closest to the ground level, the last value denotes the height furthest away.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element minOccurs="0" name="storeyHeightsBelowGround" type="gml:MeasureOrNilReasonListType">
  <annotation>
    <documentation>Lists the height of each storey below ground. The first value in the list denotes the height of the storey closest to the ground level, the last value denotes the height furthest away.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="buildingConstructiveElement">
  <annotation>
    <documentation>Relates the constructive elements to the Building or BuildingPart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="bldg:BuildingConstructiveElement"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="buildingInstallation">
  <annotation>
    <documentation>Relates the installation objects to the Building or BuildingPart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="bldg:BuildingInstallation"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="buildingRoom">
  <annotation>
    <documentation>Relates the rooms to the Building or BuildingPart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="bldg:BuildingRoom"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<sequence minOccurs="0">
  <element ref="bldg:BuildingRoom"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>
</complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="buildingFurniture">
  <annotation>
    <documentation>Relates the furniture objects to the Building or
BuildingPart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="bldg:BuildingFurniture"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="buildingSubdivision">
  <annotation>
    <documentation>Relates the logical subdivisions to the Building or
BuildingPart.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="bldg:AbstractBuildingSubdivision"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="address" type="core:AddressPropertyType">
  <annotation>
    <documentation>Relates the addresses to the Building or
BuildingPart.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractBuilding" type="bldg:ADEOfAbstractBuildingPropertyType">
  <annotation>
    <documentation>Augments AbstractBuilding with properties defined in an
200
AbstractBuildingSubdivision is the abstract superclass for different kinds of logical building subdivisions.

indicates the specific type of the building subdivision.

Specifies the intended purposes of the building subdivision.

Specifies the actual uses of the building subdivision.

Specifies qualified elevations of the building subdivision in relation to a well-defined surface which is commonly taken as origin.
(e.g. geoid or water level). [cf. INSPIRE]
<annotation>
  <documentation>Relates the rooms to the building subdivision.</documentation>
</annotation>

<element maxOccurs="unbounded" minOccurs="0"
  name="adeOfAbstractBuildingSubdivision"
  type="bldg:ADEOfAbstractBuildingSubdivisionPropertyType">
  <annotation>
    <documentation>Augments AbstractBuildingSubdivision with properties defined in an ADE.</documentation>
  </annotation>
</element>

<complexType name="AbstractBuildingSubdivisionPropertyType">
  <sequence minOccurs="0">
    <element ref="bldg:AbstractBuildingSubdivision"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="Building" substitutionGroup="bldg:AbstractBuilding" type="bldg:BuildingType">
  <annotation>
    <documentation>A Building is a free-standing, self-supporting construction that is roofed, usually walled, and can be entered by humans and is normally designed to stand permanently in one place. It is intended for human occupancy (e.g. a place of work or recreation), habitation and/or shelter of humans, animals or things.</documentation>
  </annotation>
</element>

<complexType name="BuildingType">
  <complexContent>
    <extension base="bldg:AbstractBuildingType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="buildingPart">
          <annotation>
            <documentation>Relates the building parts to the Building.</documentation>
          </annotation>
        </element>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBuilding" type="bldg:ADEOfBuildingPropertyType">
  <annotation>
    <documentation>Augments the Building with properties defined in an ADE.</documentation>
  </annotation>
</element>

<complexType name="BuildingPropertyType">
  <sequence minOccurs="0">
    <element ref="bldg:Building"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BuildingConstructiveElement" substitutionGroup="con:AbstractConstructiveElement" type="bldg:BuildingConstructiveElementType">
  <annotation>
    <documentation>A BuildingConstructiveElement is an element of a Building which is essential from a structural point of view. Examples are walls, slabs, staircases, beams.</documentation>
  </annotation>
</element>

<complexType name="BuildingConstructiveElementType">
  <complexContent>
    <extension base="con:AbstractConstructiveElementType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the BuildingConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the BuildingConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the
BuildingConstructiveElement.</documentation>
   </element>
   <element maxOccurs="unbounded" minOccurs="0"
   name="adeOfBuildingConstructiveElement"
type="bldg:ADEOfBuildingConstructiveElementPropertyType">
   <annotation>
   <documentation>Augments the BuildingConstructiveElement with properties
defined in an ADE.</documentation>
   </annotation>
   </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="BuildingConstructiveElementPropertyType">
   <sequence minOccurs="0">
   <element ref="bldg:BuildingConstructiveElement"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup"/>
   <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="BuildingFurniture" substitutionGroup="con:AbstractFurniture"
type="bldg:BuildingFurnitureType">
   <annotation>
   <documentation>A BuildingFurniture is an equipment for occupant use, usually not
fixed to the building. [cf. ISO 6707-1]</documentation>
   </annotation>
   </element>
<complexType name="BuildingFurnitureType">
   <complexContent>
   <extension base="con:AbstractFurnitureType">
   <sequence>
   <element minOccurs="0" name="class" type="gml:CodeType">
   <annotation>
   <documentation>Indicates the specific type of the
BuildingFurniture.</documentation>
   </annotation>
   </element>
   <element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
   <annotation>
   <documentation>Specifies the intended purposes of the
BuildingFurniture.</documentation>
   </annotation>
   </element>
   <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
   <annotation>
   <documentation>Specifies the actual uses of the
BuildingFurniture.</documentation>
   </annotation>
   </element>
   </sequence>
   </extension>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingFurniture" type="bldg:ADEOfBuildingFurniturePropertyType">
  <annotation>
    <documentation>Augments the BuildingFurniture with properties defined in an ADE.</documentation>
  </annotation>
</element>

<complexType name="BuildingFurniturePropertyType">
  <complexContent>
    <extension base="bldg:BuildingFurniture">
      <sequence minOccurs="0">
        <element ref="bldg:BuildingFurniture"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element name="BuildingInstallation" substitutionGroup="con:AbstractInstallation" type="bldg:BuildingInstallationType">
  <annotation>
    <documentation>A BuildingInstallation is a permanent part of a Building (inside and/or outside) which has not the significance of a BuildingPart. Examples are stairs, antennas, balconies or small roofs.</documentation>
  </annotation>
</element>

<complexType name="BuildingInstallationType">
  <complexContent>
    <extension base="con:AbstractInstallationType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the BuildingInstallation.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the BuildingInstallation.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the BuildingInstallation.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingInstallation" type="bldg:ADEOfBuildingInstallationPropertyType">
  <annotation>
    <documentation>Augments the BuildingInstallation with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="BuildingInstallationPropertyType">
  <complexContent>
    <extension base="bldg:AbstractBuildingType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingPart" type="bldg:ADEOfBuildingPartPropertyType">
          <annotation>
            <documentation>Augments the BuildingPart with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="BuildingPartPropertyType">
  <sequence minOccurs="0">
    <element ref="bldg:BuildingPart"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="BuildingRoom" substitutionGroup="core:AbstractUnoccupiedSpace" type="bldg:BuildingRoomType">
A BuildingRoom is a space within a Building or BuildingPart intended for human occupancy (e.g. a place of work or recreation) and/or containment of animals or things. A BuildingRoom is bounded physically and/or virtually (e.g. by ClosureSurfaces or GenericSurfaces).

```xml
<complexType name="BuildingRoomType">
  <complexContent>
    <extension base="core:AbstractUnoccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the BuildingRoom.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the BuildingRoom.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the BuildingRoom.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="roomHeight" type="bldg:RoomHeightPropertyType">
          <annotation>
            <documentation>Specifies qualified heights of the BuildingRoom.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="buildingFurniture">
          <annotation>
            <documentation>Relates the furniture objects to the BuildingRoom.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```
<element maxOccurs="unbounded" minOccurs="0" name="buildingInstallation">
  <annotation>
    <documentation>Relates the installation objects to the BuildingRoom.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="bldg:BuildingInstallation"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingRoom" type="bldg:ADEOfBuildingRoomPropertyType">
  <annotation>
    <documentation>Augments the BuildingRoom with properties defined in an ADE.</documentation>
  </annotation>
</element>

<complexType name="BuildingRoomPropertyType">
  <sequence minOccurs="0">
    <element ref="bldg:BuildingRoom"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="BuildingUnit" substitutionGroup="bldg:AbstractBuildingSubdivision" type="bldg:BuildingUnitType">
  <annotation>
    <documentation>A BuildingUnit is a logical subdivision of a Building. BuildingUnits are formed according to some homogeneous property like function, ownership, management, or accessibility. They may be separately sold, rented out, inherited, managed, etc.</documentation>
  </annotation>
</element>

<complexType name="BuildingUnitType">
  <complexContent>
    <extension base="bldg:AbstractBuildingSubdivisionType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="storey"
type="bldg:StoreyPropertyType">
        <documentation>Relates to the storeys on which the BuildingUnit is located.</documentation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="address" type="core:AddressPropertyType">
        <documentation>Relates to the addresses that are assigned to the BuildingUnit.</documentation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingUnit" type="bldg:ADEOfBuildingUnitPropertyType">
        <documentation>Augments the BuildingUnit with properties defined in an ADE.</documentation>
    </element>
</complexType>

<complexType name="BuildingUnitPropertyType">
    <sequence minOccurs="0">
        <element ref="bldg:BuildingUnit"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="RoomHeight" substitutionGroup="gml:AbstractObject" type="bldg:RoomHeightType">
    <documentation>The RoomHeight represents a vertical distance (measured or estimated) between a low reference and a high reference. [cf. INSPIRE]</documentation>
</element>

<complexType name="RoomHeightType">
    <sequence>
        <element name="highReference" type="gml:CodeType">
            <documentation>Indicates the high point used to calculate the value of the room height.</documentation>
        </element>
        <element name="lowReference" type="gml:CodeType">
            <documentation>Indicates the low point used to calculate the value of the room height.</documentation>
        </element>
    </sequence>
</complexType>
<element name="status" type="con:HeightStatusValueType">
    <documentation>Indicates the way the room height has been captured.</documentation>
</element>

<element name="value" type="gml:LengthType">
    <documentation>Specifies the value of the room height.</documentation>
</element>

<complexType name="RoomHeightPropertyType">
    <sequence>
        <element ref="bldg:RoomHeight"/>
    </sequence>
</complexType>

<complexType name="StoreyType">
    <complexContent>
        <extension base="bldg:AbstractBuildingSubdivisionType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="buildingUnit" type="bldg:BuildingUnitPropertyType">
                    <documentation>Relates to the building units that belong to the Storey.</documentation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfStorey" type="bldg:ADEOfStoreyPropertyType">
                    <documentation>Augments the Storey with properties defined in an ADE.</documentation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="StoreyPropertyType">
    <sequence minOccurs="0">
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfStorey" type="bldg:ADEOfStoreyPropertyType">
            <annotation>
                <documentation>Augments the Storey with properties defined in an ADE.</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
C.5. City Furniture

The CityGML City Furniture module is defined in the XML Schema Definition file cityFurniture.xsd (Listing 32). The target namespace http://www.opengis.net/citygml/cityfurniture/3.0 is associated with this module.

Listing 32. cityFurniture.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:core="http://www.opengis.net/citygml/3.0"
xmlns:frn="http://www.opengis.net/citygml/cityfurniture/3.0"
xmlns:gml="http://www.opengis.net/gml/3.2" elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/citygml/cityfurniture/3.0" version="3.0.0">
<annotation>
<documentation>The CityFurniture module supports representation of city furniture objects. City furniture objects are immovable objects like lanterns, traffic signs, advertising columns, benches, or bus stops that can be found in traffic areas, residential areas, on squares, or in built-up areas.</documentation>
</annotation>
<import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
<import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
<!--XML Schema document created by ShapeChange - http://shapechange.net/-->
<element abstract="true" name="ADEOfCityFurniture"
substitutionGroup="gml:AbstractObject" type="frn:ADEOfCityFurnitureType">
<annotation>
<documentation>ADEOfCityFurniture acts as a hook to define properties within an ADE that are to be added to a CityFurniture.</documentation>
</annotation>
</element>
<complexType abstract="true" name="ADEOfCityFurnitureType">
<sequence/>
</complexType>
<complexType name="ADEOfCityFurniturePropertyType">
<sequence>
<element ref="frn:ADEOfCityFurniture"/>
</sequence>
</complexType>
<element name="CityFurniture" substitutionGroup="core:AbstractOccupiedSpace"
type="frn:CityFurnitureType">
<annotation>
<documentation>CityFurniture is an object or piece of equipment installed in the outdoor environment for various purposes. Examples include street signs, traffic
```
signals, street lamps, benches, fountains.</documentation>
</element>
<complexType name="CityFurnitureType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the CityFurniture.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the CityFurniture.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the CityFurniture.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfCityFurniture" type="frn:ADEOfCityFurniturePropertyType">
          <annotation>
            <documentation>Augments the CityFurniture with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="CityFurniturePropertyType">
  <sequence minOccurs="0">
    <element ref="frn:CityFurniture"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>

C.6. City Object Group

The CityGML City Object Group module is defined in the XML Schema Definition file
The target namespace http://www.opengis.net/citygml/cityobjectgroup/3.0 is associated with this module.

Listing 33. cityObjectGroup.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:core="http://www.opengis.net/citygml/3.0"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:grp="http://www.opengis.net/citygml/cityobjectgroup/3.0"
    elementFormDefault="qualified"
    targetNamespace="http://www.opengis.net/citygml/cityobjectgroup/3.0" version="3.0.0">
  <annotation>
    <documentation>The CityObjectGroup module supports grouping of city objects. Arbitrary city objects may be aggregated in groups according to user-defined criteria. A group may be further classified by application-specific attributes.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2"
      schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net-->  
  <element abstract="true" name="ADEOfCityObjectGroup"
      substitutionGroup="gml:AbstractObject" type="grp:ADEOfCityObjectGroupType">
    <annotation>
      <documentation>ADEOfCityObjectGroup acts as a hook to define properties within an ADE that are to be added to a CityObjectGroup.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfCityObjectGroupType">
  </complexType>
  <complexType name="ADEOfCityObjectGroupPropertyType">
    <sequence>
      <element ref="grp:ADEOfCityObjectGroup"/>
    </sequence>
  </complexType>
  <element name="CityObjectGroup" substitutionGroup="core:AbstractLogicalSpace"
      type="grp:CityObjectGroupType">
    <annotation>
      <documentation>A CityObjectGroup represents an application-specific aggregation of city objects according to some user-defined criteria. Examples for groups are the buildings in a specific region, the result of a query, or objects put together for visualization purposes. Each member of a group may be qualified by a role name, reflecting the role each city object plays in the context of the group.</documentation>
    </annotation>
  </element>
  <complexType name="CityObjectGroupType">
    <complexContent>
      <extension base="core:AbstractLogicalSpaceType">
        <sequence>
          <element minOccurs="0" name="class" type="gml:CodeType">\n            214
```
<element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
  <annotation>
    <documentation>Specifies the intended purposes of the CityObjectGroup.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
  <annotation>
    <documentation>Specifies the actual usages of the CityObjectGroup.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="groupMember">
  <complexType>
    <complexContent>
      <extension base="gml:AbstractMemberType">
        <sequence>
          <element ref="grp:Role"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>

<element minOccurs="0" name="parent" type="gml:ReferenceType">
  <annotation>
    <documentation>Relates to a city object to which the CityObjectGroup belongs.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfCityObjectGroup"
type="grp:ADEOfCityObjectGroupPropertyType">
  <annotation>
    <documentation>Augments the CityObjectGroup with properties defined in an ADE.</documentation>
  </annotation>
</element>
C.7. Construction

The CityGML Construction module is defined in the XML Schema Definition file construction.xsd
The target namespace http://www.opengis.net/citygml/construction/3.0 is associated with this module.

**Listing 34. construction.xsd**

```xml
<?xml version="1.0" encoding="UTF-8"?
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:con="http://www.opengis.net/citygml/construction/3.0"
  xmlns:core="http://www.opengis.net/citygml/3.0"
  xmlns:gml="http://www.opengis.net/gml/3.2" elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/construction/3.0" version="3.0.0">
  <annotation>
    <documentation>The Construction module supports representation of key elements of different types of constructions. These key elements include construction surfaces (e.g. floor and ceiling), windows and doors, constructive elements (e.g. beams and slabs), installations, and furniture.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net/-->
  <element abstract="true" name="ADEOfAbstractConstruction" substitutionGroup="gml:AbstractObject" type="con:ADEOfAbstractConstructionType">
    <annotation>
      <documentation>ADEOfAbstractConstruction acts as a hook to define properties within an ADE that are to be added to AbstractConstruction.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAbstractConstructionType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractConstructionPropertyType">
    <sequence>
      <element ref="con:ADEOfAbstractConstruction"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfAbstractConstructionSurface" substitutionGroup="gml:AbstractObject" type="con:ADEOfAbstractConstructionSurfaceType">
    <annotation>
      <documentation>ADEOfAbstractConstructionSurface acts as a hook to define properties within an ADE that are to be added to AbstractConstructionSurface.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAbstractConstructionSurfaceType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractConstructionSurfacePropertyType">
    <sequence>
      <element ref="con:ADEOfAbstractConstructionSurface"/>
    </sequence>
  </complexType>
</schema>
```
<complexType abstract="true" name="ADEOfAbstractConstructiveElement"
substitutionGroup="gml:AbstractObject"
type="con:ADEOfAbstractConstructiveElementType">
  <annotation>
    <documentation>ADEOfAbstractConstructiveElement acts as a hook to define
    properties within an ADE that are to be added to
    AbstractConstructiveElement.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractConstructiveElementType">
  <sequence/>
</complexType>
<complexType name="ADEOfAbstractConstructiveElementPropertyType">
  <sequence>
    <element ref="con:ADEOfAbstractConstructiveElement"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractFillingElement"
substitutionGroup="gml:AbstractObject" type="con:ADEOfAbstractFillingElementType">
  <annotation>
    <documentation>ADEOfAbstractFillingElement acts as a hook to define properties
    within an ADE that are to be added to AbstractFillingElement.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractFillingElementType">
  <sequence/>
</complexType>
<complexType name="ADEOfAbstractFillingElementPropertyType">
  <sequence>
    <element ref="con:ADEOfAbstractFillingElement"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractFillingSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfAbstractFillingSurfaceType">
  <annotation>
    <documentation>ADEOfAbstractFillingSurface acts as a hook to define properties
    within an ADE that are to be added to AbstractFillingSurface.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractFillingSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfAbstractFillingSurfacePropertyType">
  <sequence>
    <element ref="con:ADEOfAbstractFillingSurface"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractFurniture"
substitutionGroup="gml:AbstractObject" type="con:ADEOfAbstractFurnitureType">
  <annotation>
    <documentation>ADEOfAbstractFurniture acts as a hook to define properties
    within an ADE that are to be added to AbstractFurniture.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractFurnitureType">
  <sequence/>
</complexType>
<complexType name="ADEOfAbstractFurniturePropertyType">
  <sequence>
    <element ref="con:ADEOfAbstractFurniture"/>
  </sequence>
</complexType>
<documentation>ADEOfAbstractFurniture acts as a hook to define properties within an ADE that are to be added to AbstractFurniture.</documentation>

<complexType abstract="true" name="ADEOfAbstractFurnitureType">
  <sequence/>
</complexType>

<complexType name="ADEOfAbstractFurniturePropertyType">
  <sequence>
    <element ref="con:ADEOfAbstractFurniture"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfAbstractInstallation" substitutionGroup="gml:AbstractObject" type="con:ADEOfAbstractInstallationType">
  <annotation>
    <documentation>ADEOfAbstractInstallation acts as a hook to define properties within an ADE that are to be added to AbstractInstallation.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfAbstractInstallationType">
  <sequence/>
</complexType>

<complexType name="ADEOfAbstractInstallationPropertyType">
  <sequence>
    <element ref="con:ADEOfAbstractInstallation"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfCeilingSurface" substitutionGroup="gml:AbstractObject" type="con:ADEOfCeilingSurfaceType">
  <annotation>
    <documentation>ADEOfCeilingSurface acts as a hook to define properties within an ADE that are to be added to a CeilingSurface.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfCeilingSurfaceType">
  <sequence/>
</complexType>

<complexType name="ADEOfCeilingSurfacePropertyType">
  <sequence>
    <element ref="con:ADEOfCeilingSurface"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfDoor" substitutionGroup="gml:AbstractObject" type="con:ADEOfDoorType">
  <annotation>
    <documentation>ADEOfDoor acts as a hook to define properties within an ADE that are to be added to a Door.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfDoorType">
  <sequence/>
</complexType>
<complexType name="ADEOfDoorPropertyType">
   <sequence>
      <element ref="con:ADEOfDoor"/>
   </sequence>
</complexType>

<complexType name="ADEOfDoorSurfaceType">
   <sequence/>
</complexType>

<complexType name="ADEOfDoorSurfacePropertyType">
   <sequence>
      <element ref="con:ADEOfDoorSurface"/>
   </sequence>
</complexType>

<complexType name="ADEOfFloorSurfaceType">
   <sequence/>
</complexType>

<complexType name="ADEOfFloorSurfacePropertyType">
   <sequence>
      <element ref="con:ADEOfFloorSurface"/>
   </sequence>
</complexType>

<complexType name="ADEOfGroundSurfaceType">
   <sequence/>
</complexType>

<complexType name="ADEOfGroundSurfacePropertyType">
   <sequence>
      <element ref="con:ADEOfGroundSurface"/>
   </sequence>
</complexType>

<documentation>ADEOfDoorSurface acts as a hook to define properties within an ADE that are to be added to a DoorSurface.</documentation>

<documentation>ADEOfFloorSurface acts as a hook to define properties within an ADE that are to be added to a FloorSurface.</documentation>

<documentation>ADEOfGroundSurface acts as a hook to define properties within an ADE that are to be added to a GroundSurface.</documentation>
ADEOfRoofSurface acts as a hook to define properties within an ADE that are to be added to a RoofSurface.

ADEOfWallSurface acts as a hook to define properties within an ADE that are to be added to a WallSurface.

ADEOfWindow acts as a hook to define properties within an ADE that are to be added to a Window.
<element ref="con:ADEOfWindow"/>

</complexType>
<element abstract="true" name="ADEOfWindowSurface"
substitutionGroup="gm:AbstractObject" type="con:ADEOfWindowSurfaceType">
<annotation>
    <documentation>ADEOfWindowSurface acts as a hook to define properties within an
ADE that are to be added to a WindowSurface.</documentation>
</annotation>
</element>
<complexType abstract="true" name="ADEOfWindowSurfaceType">
<sequence/>
</complexType>
<complexType name="ADEOfWindowSurfacePropertyType">
<sequence>
    <element ref="con:ADEOfWindowSurface"/>
</sequence>
</complexType>
<element abstract="true" name="AbstractConstruction"
substitutionGroup="core:AbstractOccupiedSpace" type="con:AbstractConstructionType">
<annotation>
    <documentation>AbstractConstruction is the abstract superclass for objects that
are manufactured by humans from construction materials, are connected to earth, and
are intended to be permanent. A connection with the ground also exists when the
construction rests by its own weight on the ground or is moveable limited on
stationary rails or if the construction is intended to be used mainly
stationary.</documentation>
</annotation>
</element>
<complexType abstract="true" name="AbstractConstructionType">
<complexContent>
<extension base="core:AbstractOccupiedSpaceType">
<sequence>
    <element minOccurs="0" name="conditionOfConstruction"
type="con:ConditionOfConstructionValueType">
        <annotation>
            <documentation>Indicates the life-cycle status of the construction. [cf.
INSPIRE]</documentation>
        </annotation>
    </element>
    <element minOccurs="0" name="dateOfConstruction" type="date">
        <annotation>
            <documentation>Indicates the date at which the construction was
completed.</documentation>
        </annotation>
    </element>
    <element minOccurs="0" name="dateOfDemolition" type="date">
        <annotation>
            <documentation>Indicates the date at which the construction was
demolished.</documentation>
        </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="constructionEvent" type="con:ConstructionEventPropertyType">
  <annotation>
    <documentation>Describes specific events in the life-time of the construction.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="elevation" type="con:ElevationPropertyType">
  <annotation>
    <documentation>Specifies qualified elevations of the construction in relation to a well-defined surface which is commonly taken as origin (e.g. geoid or water level). [cf. INSPIRE]</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="height" type="con:HeightPropertyType">
  <annotation>
    <documentation>Specifies qualified heights of the construction above ground or below ground. [cf. INSPIRE]</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="occupancy" type="core:OccupancyPropertyType">
  <annotation>
    <documentation>Provides qualified information on the residency of persons, animals, or other moveable objects in the construction.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractConstruction" type="con:ADEOfAbstractConstructionPropertyType">
  <annotation>
    <documentation>Augments AbstractConstruction with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</complexContent>
</complexType>
<complexType name="AbstractConstructionPropertyType">
  <sequence minOccurs="0">
    <element ref="con:AbstractConstruction"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractConstructionSurface" substitutionGroup="core:AbstractThematicSurface" type="con:AbstractConstructionSurfaceType">
AbstractConstructionSurface is the abstract superclass for different kinds of surfaces that bound a construction.

AbstractConstructionSurfaceType

fillingSurface

adeOfAbstractConstructionSurface

AbstractConstrictiveElement

AbstractConstrictiveElementType

isStructuralElement

AbstractConstrictiveElementType

isStructuralElement
from a structural point of view. A structural element cannot be omitted without collapsing of the construction. Examples are pylons and anchorages of bridges.
AbstractFillingSurface is the abstract superclass for different kinds of surfaces that seal openings filled by filling elements.

AbstractFillingSurfaceType

AbstractFurniture is the abstract superclass for the representation of furniture objects of a construction.

AbstractFurnitureType
type="con:ADEOfAbstractFurniturePropertyType">
  <annotation>
    <documentation>Augments AbstractFurniture with properties defined in an
    ADE.</documentation>
  </annotation>
  </element>
</complexType>
<complexType name="AbstractFurniturePropertyType">
  <sequence minOccurs="0">
    <element ref="con:AbstractFurniture"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractInstallation" substitutionGroup="core:AbstractOccupiedSpace" type="con:AbstractInstallationType">
  <annotation>
    <documentation>AbstractInstallation is the abstract superclass for the
    representation of installation objects of a construction.</documentation>
  </annotation>
  </element>
<complexType abstract="true" name="AbstractInstallationType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="relationToConstruction" type="con:RelationToConstructionType">
          <annotation>
            <documentation>Indicates whether the installation is located inside
            and/or outside of the construction.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractInstallation" type="con:ADEOfAbstractInstallationPropertyType">
          <annotation>
            <documentation>Augments AbstractInstallation with properties defined in
            an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AbstractInstallationPropertyType">
  <sequence minOccurs="0">
    <element ref="con:AbstractInstallation"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
<complexType name="CeilingSurfaceType">
<complexContent>
<extension base="con:AbstractConstructionSurfaceType">
<sequence>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfCeilingSurface" type="con:ADEOfCeilingSurfacePropertyType">
<annotation>
<documentation>Augments the CeilingSurface with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="CeilingSurfacePropertyType">
<sequence minOccurs="0">
<element ref="con:CeilingSurface"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<simpleType name="ConditionOfConstructionValueType">
<annotation>
<documentation>ConditionOfConstructionValue enumerates different conditions of a construction. [cf. INSPIRE]</documentation>
</annotation>
<restriction base="string">
<enumeration value="declined">
<annotation>
<documentation>Indicates that the construction cannot be used under normal conditions, though its main elements (walls, roof) are still present. [cf. INSPIRE]</documentation>
</annotation>
</enumeration>
<enumeration value="demolished">
<annotation>
<documentation>Indicates that the construction has been demolished. There are no more visible remains. [cf. INSPIRE]</documentation>
</annotation>
</enumeration>
<enumeration value="functional">
</enumeration>
</restriction>
</simpleType>
<annotation>
  <documentation>Indicates that the construction is functional. [cf. INSPIRE]</documentation>
</annotation>
</enumeration>
<enumeration value="projected">
  <annotation>
    <documentation>Indicates that the construction is being designed. Construction works have not yet started. [cf. INSPIRE]</documentation>
  </annotation>
</enumeration>
<enumeration value="ruin">
  <annotation>
    <documentation>Indicates that the construction has been partly demolished and some main elements (roof, walls) have been destroyed. There are some visible remains of the construction. [cf. INSPIRE]</documentation>
  </annotation>
</enumeration>
<enumeration value="underConstruction">
  <annotation>
    <documentation>Indicates that the construction is under construction and not yet functional. This applies only to the initial construction works of the construction and not to maintenance work. [cf. INSPIRE]</documentation>
  </annotation>
</enumeration>
</restriction>
</simpleType>
<element name="ConstructionEvent" substitutionGroup="gml:AbstractObject" type="con:ConstructionEventType">
  <annotation>
    <documentation>A ConstructionEvent is a data type used to describe a specific event that is associated with a construction. Examples are the issuing of a building permit or the renovation of a building.</documentation>
  </annotation>
</element>
<complexType name="ConstructionEventType">
  <sequence>
    <element name="event" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the specific event type.</documentation>
      </annotation>
    </element>
    <element name="dateOfEvent" type="date">
      <annotation>
        <documentation>Specifies the date at which the event took or will take place.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="description" type="string">
      <annotation>
        <documentation>Provides additional information on the event.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="ConstructionEventPropertyType">
  <sequence>
    <element ref="con:ConstructionEvent"/>
  </sequence>
</complexType>

<element name="Door" substitutionGroup="con:AbstractFillingElement" type="con:DoorType">
  <annotation>
    <documentation>A Door is a construction for closing an opening intended primarily for access or egress or both. [cf. ISO 6707-1]</documentation>
  </annotation>
</element>

<complexType name="DoorType">
  <complexContent>
    <extension base="con:AbstractFillingElementType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the Door.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the Door.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the Door.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="address" type="core:AddressPropertyType">
          <annotation>
            <documentation>Relates to the addresses that are assigned to the Door.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfDoor" type="con:ADEOfDoorPropertyType">
          <annotation>
            <documentation>Augments the Door with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="DoorPropertyType">
  <sequence minOccurs="0">
    <element ref="con:Door"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="DoorSurface" substitutionGroup="con:AbstractFillingSurface" type="con:DoorSurfaceType">
  <annotation>
    <documentation>A DoorSurface is either a boundary surface of a Door feature or a surface that seals an opening filled by a door.</documentation>
  </annotation>
</element>

<complexType name="DoorSurfaceType">
  <complexContent>
    <extension base="con:AbstractFillingSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="address" type="core:AddressPropertyType">
          <annotation>
            <documentation>Relates to the addresses that are assigned to the DoorSurface.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfDoorSurface" type="con:ADEOfDoorSurfacePropertyType">
          <annotation>
            <documentation>Augments the DoorSurface with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="DoorSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:DoorSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="Elevation" substitutionGroup="gml:AbstractObject" type="con:ElevationType">
  <annotation>
Elevation is a data type that includes the elevation value itself and information on how this elevation was measured. [cf. INSPIRE]

<complexType name="ElevationType">
  <sequence>
    <element name="elevationReference" type="gml:CodeType">
      <annotation>
        Specifies the level from which the elevation was measured. [cf. INSPIRE]
      </annotation>
    </element>
    <element name="elevationValue" type="gml:DirectPositionType">
      <annotation>
        Specifies the value of the elevation. [cf. INSPIRE]
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="ElevationPropertyType">
  <sequence>
    <element ref="con:Elevation"/>
  </sequence>
</complexType>

<element name="FloorSurface" substitutionGroup="con:AbstractConstructionSurface" type="con:FloorSurfaceType">
  <annotation>
    A FloorSurface is surface that represents the interior floor of a construction. An example is the floor of a room.
  </annotation>
</element>

<complexType name="FloorSurfaceType">
  <complexContent>
    <extension base="con:AbstractConstructionSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfFloorSurface" type="con:ADEOfFloorSurfacePropertyType">
          <annotation>
            Augments the FloorSurface with properties defined in an ADE.
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="FloorSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:FloorSurface"/>
  </sequence>
</complexType>
<complexType name="GroundSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:GroundSurface"/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </sequence>
</complexType>

<element name="Height" substitutionGroup="gml:AbstractObject" type="con:HeightType">
  <annotation>
    <documentation>Height represents a vertical distance (measured or estimated)
    between a low reference and a high reference. [cf. INSPIRE]</documentation>
  </annotation>
</element>

<complexType name="HeightType">
  <sequence>
    <element name="highReference" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the high point used to calculate the value of the
        height. [cf. INSPIRE]</documentation>
      </annotation>
    </element>
    <element name="lowReference" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the low point used to calculate the value of the
        height. [cf. INSPIRE]</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="HeightPropertyType">
  <sequence>
    <element ref="con:Height"/>
  </sequence>
</complexType>

<complexType name="HeightStatusValueType">
  <annotation>
    <documentation>HeightStatusValue enumerates the different methods used to capture a height. [cf. INSPIRE]</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="estimated">
      <annotation>
        <documentation>Indicates that the height has been estimated and not measured. [cf. INSPIRE]</documentation>
      </annotation>
    </enumeration>
    <enumeration value="measured">
      <annotation>
        <documentation>Indicates that the height has been (directly or indirectly) measured. [cf. INSPIRE]</documentation>
      </annotation>
    </enumeration>
  </restriction>
</complexType>

<element name="InteriorWallSurface" substitutionGroup="con:AbstractConstructionSurface" type="con:InteriorWallSurfaceType">
  <annotation>
    <documentation>An InteriorWallSurface is a surface that is visible from inside a construction. An example is the wall of a room.</documentation>
  </annotation>
</element>

<complexType name="InteriorWallSurfaceType">
  <complexContent>
    <complexType>
      <sequence>
        <element name="status" type="con:HeightStatusValueType">
          <annotation>
            <documentation>Indicates the way the height has been captured. [cf. INSPIRE]</documentation>
          </annotation>
        </element>
        <element name="value" type="gml:LengthType">
          <annotation>
            <documentation>Specifies the value of the height above or below ground. [cf. INSPIRE]</documentation>
          </annotation>
        </element>
      </sequence>
    </complexType>
  </complexContent>
</complexType>
<extension base="con:AbstractConstructionSurfaceType">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="0" name="adeOfInteriorWallSurface" type="con:ADEOfInteriorWallSurfacePropertyType">
      <annotation>
        <documentation>Augments the InteriorWallSurface with properties defined in an ADE.</documentation>
      </annotation>
      </element>
  </sequence>
</extension>

<complexType name="InteriorWallSurfacePropertyType">
  <complexContent>
    <extension base="con:AbstractConstructionSurfaceType">
      <sequence minOccurs="0">
        <element ref="con:InteriorWallSurface"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element name="OtherConstruction" substitutionGroup="con:AbstractConstruction" type="con:OtherConstructionType">
  <annotation>
    <documentation>An OtherConstruction is a construction that is not covered by any of the other subclasses of AbstractConstruction.</documentation>
  </annotation>
</element>

<complexType name="OtherConstructionType">
  <complexContent>
    <extension base="con:AbstractConstructionType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the OtherConstruction.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the OtherConstruction.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the OtherConstruction.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfOtherConstruction" type="con:ADEOfOtherConstructionPropertyType">
  <annotation>
    <documentation>Augments the OtherConstruction with properties defined in an ADE.</documentation>
  </annotation>
  <element/>
</extension>
</complexType>
<complexType name="OtherConstructionPropertyType">
  <complexContent>
    <sequence minOccurs="0">
      <element ref="con:OtherConstruction"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </complexContent>
</complexType>
<element name="OuterCeilingSurface" substitutionGroup="con:AbstractConstructionSurface" type="con:OuterCeilingSurfaceType">
  <annotation>
    <documentation>An OuterCeilingSurface is a surface that belongs to the outer building shell with the orientation pointing downwards. An example is the ceiling of a loggia.</documentation>
  </annotation>
  <element/>
</complexType>
<complexType name="OuterCeilingSurfaceType">
  <complexContent>
    <extension base="con:AbstractConstructionSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfOuterCeilingSurface" type="con:ADEOfOuterCeilingSurfacePropertyType">
          <annotation>
            <documentation>Augments the OuterCeilingSurface with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="OuterFloorSurface" substitutionGroup="con:AbstractConstructionSurface" type="con:OuterFloorSurfaceType">
An OuterFloorSurface is a surface that belongs to the outer construction shell with the orientation pointing upwards. An example is the floor of a loggia.

Augments the OuterFloorSurface with properties defined in an ADE.
<element name="RoofSurface" substitutionGroup="con:AbstractConstructionSurface" type="con:RoofSurfaceType">
  <annotation>
    <documentation>A RoofSurface is a surface that delimits major roof parts of a construction.</documentation>
  </annotation>
</element>
<complexType name="RoofSurfaceType">
  <complexContent>
    <extension base="con:AbstractConstructionSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfRoofSurface" type="con:ADEOfRoofSurfacePropertyType">
          <annotation>
            <documentation>Augments the RoofSurface with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="RoofSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:RoofSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="WallSurface" substitutionGroup="con:AbstractConstructionSurface" type="con:WallSurfaceType">
  <annotation>
    <documentation>A WallSurface is a surface that is part of the building facade visible from the outside.</documentation>
  </annotation>
</element>
<complexType name="WallSurfaceType">
  <complexContent>
    <extension base="con:AbstractConstructionSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfWallSurface" type="con:ADEOfWallSurfacePropertyType">
          <annotation>
            <documentation>Augments the WallSurface with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="WallSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:WallSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="Window" substitutionGroup="con:AbstractFillingElement" type="con:WindowType">
  <annotation>
    <documentation>A Window is a construction for closing an opening in a wall or roof, primarily intended to admit light and/or provide ventilation. [cf. ISO 6707-1]</documentation>
  </annotation>
</element>

<complexType name="WindowType">
<complexContent>
<extension base="con:AbstractFillingElementType">
  <sequence>
    <element minOccurs="0" name="class" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the specific type of the Window.</documentation>
      </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
      <annotation>
        <documentation>Specifies the intended purposes of the Window.</documentation>
      </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
      <annotation>
        <documentation>Specifies the actual uses of the Window.</documentation>
      </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="adeOfWindow" type="con:ADEOfWindowPropertyType">
      <annotation>
        <documentation>Augments the Window with properties defined in an ADE.</documentation>
      </annotation>
    </element>
  </sequence>
</extension>
</complexContent>
</complexType>
C.8. Dynamizer

The CityGML Dynamizer module is defined in the XML Schema Definition file dynamizer.xsd (Listing 35). The target namespace http://www.opengis.net/citygml/dynamizer/3.0 is associated with this module.

Listing 35. dynamizer.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:core="http://www.opengis.net/citygml/3.0"
        xmlns:dyn="http://www.opengis.net/citygml/dynamizer/3.0">
```
The Dynamizer module supports the injection of timeseries data for individual attributes of CityGML features. Timeseries data can either be retrieved from external Sensor APIs (e.g. OGC SensorThings API, OGC Sensor Observation Services, MQTT, proprietary platforms), external standardized timeseries files (e.g. OGC TimeseriesML or OGC Observations & Measurements), external tabulated files (e.g. CSV) or can be represented inline as basic time-value pairs.
<complexType abstract="true" name="ADEOfCompositeTimeseriesType">
    <sequence/>
</complexType>
<complexType name="ADEOfCompositeTimeseriesPropertyType">
    <sequence>
        <element ref="dyn:ADEOfCompositeTimeseries"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfDynamizer"
    substitutionGroup="gml:AbstractObject" type="dyn:ADEOfDynamizerType">
    <annotation>
        <documentation>ADEOfDynamizer acts as a hook to define properties within an ADE that are to be added to a Dynamizer.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfDynamizerType">
    <sequence/>
</complexType>
<complexType name="ADEOfDynamizerPropertyType">
    <sequence>
        <element ref="dyn:ADEOfDynamizer"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfGenericTimeseries"
    substitutionGroup="gml:AbstractObject" type="dyn:ADEOfGenericTimeseriesType">
    <annotation>
        <documentation>ADEOfGenericTimeseries acts as a hook to define properties within an ADE that are to be added to a GenericTimeseries.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfGenericTimeseriesType">
    <sequence/>
</complexType>
<complexType name="ADEOfGenericTimeseriesPropertyType">
    <sequence>
        <element ref="dyn:ADEOfGenericTimeseries"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfStandardFileTimeseries"
    substitutionGroup="gml:AbstractObject" type="dyn:ADEOfStandardFileTimeseriesType">
    <annotation>
        <documentation>ADEOfStandardFileTimeseries acts as a hook to define properties within an ADE that are to be added to a StandardFileTimeseries.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfStandardFileTimeseriesType">
    <sequence/>
</complexType>
<complexType name="ADEOfStandardFileTimeseriesPropertyType">
    <sequence>
        <element ref="dyn:ADEOfStandardFileTimeseries"/>
    </sequence>
</complexType>
<complexType abstract="true" name="ADEOfTabulatedFileTimeseries"
substitutionGroup="gml:AbstractObject" type="dyn:ADEOfTabulatedFileTimeseriesType">
  <annotation>
    <documentation>ADEOfTabulatedFileTimeseries acts as a hook to define properties within an ADE that are to be added to a TabulatedFileTimeseries.</documentation>
  </annotation>
</complexType>
<complexType abstract="true" name="ADEOfTabulatedFileTimeseriesType">
</complexType>
<complexType name="ADEOfTabulatedFileTimeseriesPropertyType">
  <sequence>
    <element ref="dyn:ADEOfTabulatedFileTimeseries"/>
  </sequence>
</complexType>
<element abstract="true" name="AbstractAtomicTimeseries"
substitutionGroup="dyn:AbstractTimeseries" type="dyn:AbstractAtomicTimeseriesType">
  <annotation>
    <documentation>AbstractAtomicTimeseries represents the attributes and relationships that are common to all kinds of atomic timeseries (GenericTimeseries, TabulatedFileTimeseries, StandardFileTimeseries). An atomic timeseries represents time-varying data of a specific data type for a single contiguous time interval.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractAtomicTimeseriesType">
  <complexContent>
    <extension base="dyn:AbstractTimeseriesType">
      <sequence>
        <element name="observationProperty" type="string">
          <annotation>
            <documentation>Specifies the phenomenon for which the atomic timeseries provides observation values.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="uom" type="string">
          <annotation>
            <documentation>Specifies the unit of measurement of the observation values.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" maxOccurs="unbounded">
          <element name="adeOfAbstractAtomicTimeseries" type="dyn:ADEOfAbstractAtomicTimeseriesPropertyType">
            <annotation>
              <documentation>Augments AbstractAtomicTimeseries with properties defined in an ADE.</documentation>
            </annotation>
          </element>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AbstractAtomicTimeseriesPropertyType">
  <sequence minOccurs="0">
    <element ref="dyn:AbstractAtomicTimeseries"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<complexType name="AbstractTimeseriesPropertyType">
  <sequence minOccurs="0">
    <element ref="dyn:AbstractTimeseries"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<complexType name="AbstractTimeseriesType">
  <complexContent>
    <extension base="core:AbstractFeatureType">
      <sequence>
        <element minOccurs="0" name="firstTimestamp" type="gml:TimePositionType">
          <annotation>
            <documentation>Specifies the beginning of the timeseries.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="lastTimestamp" type="gml:TimePositionType">
          <annotation>
            <documentation>Specifies the end of the timeseries.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractTimeseries" type="dyn:ADEOfAbstractTimeseriesPropertyType">
          <annotation>
            <documentation>Augments AbstractTimeseries with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element abstract="true" name="AbstractTimeseries" substitutionGroup="core:AbstractFeature" type="dyn:AbstractTimeseriesType">
  <annotation>
    <documentation>AbstractTimeseries is the abstract superclass representing any type of timeseries data.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractTimeseriesType">
  <complexContent>
    <extension base="core:AbstractFeatureType">
      <sequence>
        <element minOccurs="0" name="firstTimestamp" type="gml:TimePositionType">
          <annotation>
            <documentation>Specifies the beginning of the timeseries.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="lastTimestamp" type="gml:TimePositionType">
          <annotation>
            <documentation>Specifies the end of the timeseries.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractTimeseries" type="dyn:ADEOfAbstractTimeseriesPropertyType">
          <annotation>
            <documentation>Augments AbstractTimeseries with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
A CompositeTimeseries is a (possibly recursive) aggregation of atomic and composite timeseries. The components of a composite timeseries must have non-overlapping time intervals.

Relates to the atomic and composite timeseries that are part of the CompositeTimeseries. The referenced timeseries are sequentially ordered.

Augments the CompositeTimeseries with properties defined in an ADE.

A Dynamizer is an object that injects timeseries data for an individual attribute of the city object in which it is included. The timeseries data overrides the static value of the referenced city object attribute in order to represent dynamic (time-dependent) variations of its value.
<element name="attributeRef" type="string">
  <annotation>
    <documentation>Specifies the attribute of a CityGML feature whose value is overridden or replaced by the (dynamic) values specified by the Dynamizer.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="startTime" type="gml:TimePositionType">
  <annotation>
    <documentation>Specifies the beginning of the time span for which the Dynamizer provides dynamic values.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="endTime" type="gml:TimePositionType">
  <annotation>
    <documentation>Specifies the end of the time span for which the Dynamizer provides dynamic values.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="dynamicData">
  <annotation>
    <documentation>Relates to the timeseries data that is given either inline within a CityGML dataset or by a link to an external file containing timeseries data.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="dyn:AbstractTimeseries"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element minOccurs="0" name="sensorConnection" type="dyn:SensorConnectionPropertyType">
  <annotation>
    <documentation>Relates to the sensor API that delivers timeseries data.</documentation>
  </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfDynamizer" type="dyn:ADEOfDynamizerPropertyType">
  <annotation>
    <documentation>Augments the Dynamizer with properties defined in an ADE.</documentation>
  </annotation>
</element>
<complexType name="DynamizerPropertyType">
  <sequence minOccurs="0">
    <element ref="dyn:Dynamizer"/>
  </sequence>
</complexType>
<complexType name="GenericTimeseriesType">
  <complexContent>
    <extension base="dyn:AbstractAtomicTimeseriesType">
      <sequence>
        <element name="valueType" type="dyn:TimeseriesTypeValueType">
          <annotation>
            <documentation>Indicates the specific type of all time-value-pairs that are part of the GenericTimeseries.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" name="timeValuePair" type="dyn:TimeValuePairPropertyType">
          <annotation>
            <documentation>Relates to the time-value-pairs that are part of the GenericTimeseries.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfGenericTimeseries" type="dyn:ADEOfGenericTimeseriesPropertyType">
          <annotation>
            <documentation>Augments the GenericTimeseries with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="GenericTimeseriesPropertyType">
  <sequence minOccurs="0">
    <element ref="dyn:GenericTimeseries"/>
  </sequence>
</complexType>
<element name="SensorConnection" substitutionGroup="gml:AbstractObject" type="dyn:SensorConnectionType">
  <annotation>
    <documentation>A SensorConnection provides all details that are required to retrieve a specific datastream from an external sensor web service. This data type comprises the service type (e.g. OGC SensorThings API, OGC Sensor Observation Services, MQTT, proprietary platforms), the URL of the sensor service, the identifier for the sensor or thing, and its observed property as well as information about the required authentication method.</documentation>
  </annotation>
</element>

<complexType name="SensorConnectionType">
  <sequence>
    <element name="connectionType" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the type of Sensor API to which the SensorConnection refers.</documentation>
      </annotation>
    </element>
    <element name="observationProperty" type="string">
      <annotation>
        <documentation>Specifies the phenomenon for which the SensorConnection provides observations.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="uom" type="string">
      <annotation>
        <documentation>Specifies the unit of measurement of the observations.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="sensorID" type="string">
      <annotation>
        <documentation>Specifies the unique identifier of the sensor from which the SensorConnection retrieves observations.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="sensorName" type="string">
      <annotation>
        <documentation>Specifies the name of the sensor from which the SensorConnection retrieves observations.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="observationID" type="string">
      <annotation>
        <documentation>Specifies the unique identifier of the observation that is retrieved by the SensorConnection.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<element minOccurs="0" name="datastreamID" type="string">
  <annotation>
    <documentation>Specifies the datastream that is retrieved by the SensorConnection.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="baseURL" type="anyURI">
  <annotation>
    <documentation>Specifies the base URL of the Sensor API request.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="authType" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the type of authentication required to be able to access the Sensor API.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="mqttServer" type="string">
  <annotation>
    <documentation>Specifies the name of the MQTT Server. This attribute is relevant when the MQTT Protocol is used to connect to a Sensor API.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="mqttTopic" type="string">
  <annotation>
    <documentation>Names the specific datastream that is retrieved by the SensorConnection. This attribute is relevant when the MQTT Protocol is used to connect to a Sensor API.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="linkToObservation" type="string">
  <annotation>
    <documentation>Specifies the complete URL to the observation request.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="linkToSensorDescription" type="string">
  <annotation>
    <documentation>Specifies the complete URL to the sensor description request.</documentation>
  </annotation>
</element>

<element minOccurs="0" name="sensorLocation" type="gml:ReferenceType">
  <annotation>
    <documentation>Relates the sensor to the city object where it is located.</documentation>
  </annotation>
</element>

<appinfo>
  <targetElement xmlns="http://www.opengis.net/gml/3.2">core:AbstractCityObject</targetElement>
</appinfo>
<complexType name="SensorConnectionPropertyType">
  <sequence>
    <element ref="dyn:SensorConnection"/>
  </sequence>
</complexType>

<element name="StandardFileTimeseries" substitutionGroup="dyn:AbstractAtomicTimeseries" type="dyn:StandardFileTimeseriesType">
  <annotation>
    <documentation>A StandardFileTimeseries represents time-varying data for a single contiguous time interval. The data is provided in an external file referenced in the StandardFileTimeseries. The data within the external file is encoded according to a dedicated format for the representation of timeseries data such as using the OGC TimeseriesML or OGC Observations &amp; Measurements Standard. The data type of the data has to be specified within the external file.</documentation>
  </annotation>
</element>

<complexType name="StandardFileTimeseriesType">
  <complexContent>
    <extension base="dyn:AbstractAtomicTimeseriesType">
      <sequence>
        <element name="fileLocation" type="anyURI">
          <annotation>
            <documentation>Specifies the URI that points to the external timeseries file.</documentation>
          </annotation>
        </element>
        <element name="fileType" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the format used to represent the timeseries data.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="mimeType" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the MIME type of the external timeseries file.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="adeOfStandardFileTimeseries" type="dyn:ADEOfStandardFileTimeseriesPropertyType">
          <annotation>
            <documentation>Augments the StandardFileTimeseries with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
A TabulatedFileTimeseries represents time-varying data of a specific data type for a single contiguous time interval. The data is provided in an external file referenced in the TabulatedFileTimeseries. The file contains table structured data using an appropriate file format such as comma-separated values (CSV), Microsoft Excel (XLSX) or Google Spreadsheet. The timestamps and the values are given in specific columns of the table. Each row represents a single time-value-pair. A subset of rows can be selected using the idColumn and idValue attributes.
<element minOccurs="0" name="numberOfHeaderLines" type="integer">
<annotation>
<documentation>Indicates the number of lines at the beginning of the tabulated file that represent headers.</documentation>
</annotation>
</element>

<element name="fieldSeparator" type="string">
<annotation>
<documentation>Indicates which symbol is used to separate the individual values in the tabulated file.</documentation>
</annotation>
</element>

<element minOccurs="0" name="decimalSymbol" type="string">
<annotation>
<documentation>Indicates which symbol is used to separate the integer part from the fractional part of a decimal number.</documentation>
</annotation>
</element>

<element minOccurs="0" name="idColumnNo" type="integer">
<annotation>
<documentation>Specifies the number of the column that stores the identifier of the time-value-pair.</documentation>
</annotation>
</element>

<element minOccurs="0" name="idColumnName" type="string">
<annotation>
<documentation>Specifies the name of the column that stores the identifier of the time-value-pair.</documentation>
</annotation>
</element>

<element minOccurs="0" name="idValue" type="string">
<annotation>
<documentation>Specifies the value of the identifier for which the time-value-pairs are to be selected.</documentation>
</annotation>
</element>

<element minOccurs="0" name="timeColumnNo" type="integer">
<annotation>
<documentation>Specifies the number of the column that stores the timestamp of the time-value-pair.</documentation>
</annotation>
</element>

<element minOccurs="0" name="timeColumnName" type="string">
<annotation>
<documentation>Specifies the name of the column that stores the timestamp of the time-value-pair.</documentation>
</annotation>
</element>

<element minOccurs="0" name="valueColumnNo" type="integer">
<annotation>
</annotation>
</element>
<documentation>Specifies the number of the column that stores the value of the time-value-pair.</documentation>
</annotation>

<element minOccurs="0" name="valueColumnName" type="string">
<annotation>
<documentation>Specifies the name of the column that stores the value of the time-value-pair.</documentation>
</annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfTabulatedFileTimeseries" type="dyn:ADEOfTabulatedFileTimeseriesPropertyType">
<annotation>
<documentation>Augments the TabulatedFileTimeseries with properties defined in an ADE.</documentation>
</annotation>
</element>

</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TabulatedFileTimeseriesPropertyType">
<sequence minOccurs="0">
<element ref="dyn:TabulatedFileTimeseries"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TimeValuePair" substitutionGroup="gml:AbstractObject" type="dyn:TimeValuePairType">
<annotation>
<documentation>A TimeValuePair represents a value that is valid for a given timepoint. For each TimeValuePair, only one of the value properties can be used mutually exclusive. Which value property has to be provided depends on the selected value type in the GenericTimeSeries feature, in which the TimeValuePair is included.</documentation>
</annotation>
</element>
<complexType name="TimeValuePairType">
<sequence>
<element name="timestamp" type="gml:TimePositionType">
<annotation>
<documentation>Specifies the timepoint at which the value of the TimeValuePair is valid.</documentation>
</annotation>
</element>
<element minOccurs="0" name="intValue" type="integer">
<annotation>
<documentation>Specifies the "Integer" value of the TimeValuePair.</documentation>
</annotation>
</element>
</sequence>
</complexType>
<documentation>Specifies the "Double" value of the TimeValuePair.</documentation>
</annotation>
</element>
<element minOccurs="0" name="stringValue" type="string">
<annotation>
<documentation>Specifies the "String" value of the TimeValuePair.</documentation>
</annotation>
</element>
<element minOccurs="0" name="geometryValue" type="gml:GeometryPropertyType">
<annotation>
<documentation>Specifies the geometry value of the TimeValuePair.</documentation>
</annotation>
</element>
<element minOccurs="0" name="uriValue" type="anyURI">
<annotation>
<documentation>Specifies the "URI" value of the TimeValuePair.</documentation>
</annotation>
</element>
<element minOccurs="0" name="boolValue" type="boolean">
<annotation>
<documentation>Specifies the "Boolean" value of the TimeValuePair.</documentation>
</annotation>
</element>
<element minOccurs="0" name="implicitGeometryValue">
<annotation>
<documentation>Specifies the "ImplicitGeometry" value of the TimeValuePair.</documentation>
</annotation>
</element>
<element minOccurs="0" name="appearanceValue">
<annotation>
<documentation>Specifies the "Appearance" value of the
<complexType name="TimeValuePairPropertyType">
  <sequence>
    <element ref="dyn:TimeValuePair"/>
  </sequence>
</complexType>

<complexType name="TimeseriesComponentType">
  <sequence>
    <element name="repetitions" type="integer">
      <annotation>
        <documentation>Specifies how often the timeseries that is referenced by the TimeseriesComponent should be iterated.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="additionalGap" type="duration">
      <annotation>
        <documentation>Specifies how much extra time is added after all repetitions as an additional gap.</documentation>
      </annotation>
    </element>
    <element name="timeseries" type="dyn:AbstractTimeseriesPropertyType">
      <annotation>
        <documentation>Relates a timeseries to the TimeseriesComponent.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="TimeseriesComponentPropertyType">
  <sequence>
  </sequence>
</complexType>
<element ref="dyn:TimeseriesComponent"/>
</sequence>
</complexType>
<simpleType name="TimeseriesTypeValueType">
<annotation>
<documentation>TimeseriesTypeValue enumerates the possible value types for
GenericTimeseries and TimeValuePair.</documentation>
</annotation>
<restriction base="string">
<enumeration value="int">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are of
type "Integer".</documentation>
</annotation>
</enumeration>
<enumeration value="double">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are of
type "Double".</documentation>
</annotation>
</enumeration>
<enumeration value="string">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are of
type "String".</documentation>
</annotation>
</enumeration>
<enumeration value="geometry">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are
generics.</documentation>
</annotation>
</enumeration>
<enumeration value="uri">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are of
type "URI".</documentation>
</annotation>
</enumeration>
<enumeration value="bool">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are of
type "Boolean".</documentation>
</annotation>
</enumeration>
<enumeration value="implicitGeometry">
<annotation>
<documentation>Indicates that the values of the GenericTimeseries are of
type "ImplicitGeometry".</documentation>
</annotation>
</enumeration>
</restriction>
</simpleType>
C.9. Generics

The CityGML Generics module is defined in the XML Schema Definition file generics.xsd (Listing 36). The target namespace http://www.opengis.net/citygml/generics/3.0 is associated with this module.

Listing 36. generics.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:core="http://www.opengis.net/citygml/3.0"
xmlns:gen="http://www.opengis.net/citygml/generics/3.0"
xmlns:gml="http://www.opengis.net/gml/3.2" elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/citygml/generics/3.0" version="3.0.0">
  <annotation>
    <documentation>The Generics module supports application-specific extensions to the CityGML conceptual model. These extensions may be used to model and exchange additional attributes and features not covered by the predefined thematic classes of CityGML. Generic extensions shall only be used if appropriate thematic classes or attributes are not provided by any other CityGML module.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net/-->
  <element abstract="true" name="ADEOfGenericLogicalSpace"
substitutionGroup="gml:AbstractObject" type="gen:ADEOfGenericLogicalSpaceType">
    <annotation>
      <documentation>ADEOfGenericLogicalSpace acts as a hook to define properties within an ADE that are to be added to a GenericLogicalSpace.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfGenericLogicalSpaceType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfGenericLogicalSpacePropertyType">
    <sequence>
      <element ref="gen:ADEOfGenericLogicalSpace"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfGenericOccupiedSpace"
substitutionGroup="gml:AbstractObject" type="gen:ADEOfGenericOccupiedSpaceType">
    <annotation>
      <documentation>Indicates that the values of the GenericTimeseries are of type "Appearance".</documentation>
    </annotation>
  </element>
</schema>
```
<complexType abstract="true" name="ADEOfGenericOccupiedSpaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfGenericOccupiedSpacePropertyType">
  <sequence>
    <element ref="gen:ADEOfGenericOccupiedSpace"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfGenericOccupiedSpace" substitutionGroup="gml:AbstractObject" type="gen:ADEOfGenericOccupiedSpaceType">
  <annotation>
    <documentation>ADEOfGenericOccupiedSpace acts as a hook to define properties within an ADE that are to be added to a GenericOccupiedSpace.</documentation>
  </annotation>
</element>
</complexType>
<complexType name="ADEOfGenericThematicSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfGenericThematicSurfacePropertyType">
  <sequence>
    <element ref="gen:ADEOfGenericThematicSurface"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfGenericThematicSurface" substitutionGroup="gml:AbstractObject" type="gen:ADEOfGenericThematicSurfaceType">
  <annotation>
    <documentation>ADEOfGenericThematicSurface acts as a hook to define properties within an ADE that are to be added to a GenericThematicSurface.</documentation>
  </annotation>
</element>
</complexType>
<complexType name="ADEOfGenericUnoccupiedSpaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfGenericUnoccupiedSpacePropertyType">
  <sequence>
    <element ref="gen:ADEOfGenericUnoccupiedSpace"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfGenericUnoccupiedSpace" substitutionGroup="gml:AbstractObject" type="gen:ADEOfGenericUnoccupiedSpaceType">
  <annotation>
    <documentation>ADEOfGenericUnoccupiedSpace acts as a hook to define properties within an ADE that are to be added to a GenericUnoccupiedSpace.</documentation>
  </annotation>
</element>
</complexType>
<complexType name="CodeAttributeType">
  <sequence/>
</complexType>
<element name="CodeAttribute" substitutionGroup="core:AbstractGenericAttribute" type="gen:CodeAttributeType">
  <annotation>
    <documentation>CodeAttribute is a data type used to define generic attributes of type "Code".</documentation>
  </annotation>
</element>
</complexType>
<complexContent>
  <extension base="core:AbstractGenericAttributeType">
    <sequence>
      <element name="name" type="string">
        <annotation>
          <documentation>Specifies the name of the CodeAttribute.</documentation>
        </annotation>
      </element>
      <element name="value" type="gml:CodeType">
        <annotation>
          <documentation>Specifies the "Code" value.</documentation>
        </annotation>
      </element>
    </sequence>
  </extension>
</complexContent>

<complexType name="CodeAttributePropertyType">
  <sequence>
    <element ref="gen:CodeAttribute"/>
  </sequence>
</complexType>

<complexType name="DateAttributeType">
  <complexContent>
    <extension base="core:AbstractGenericAttributeType">
      <sequence>
        <element name="name" type="string">
          <annotation>
            <documentation>Specifies the name of the DateAttribute.</documentation>
          </annotation>
        </element>
        <element name="value" type="date">
          <annotation>
            <documentation>Specifies the "Date" value.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="DateAttributePropertyType">
  <sequence>
    <element ref="gen:DateAttribute"/>
  </sequence>
</complexType>
<complexType name="DoubleAttributePropertyType">
  <sequence>
    <element ref="gen:DoubleAttribute"/>
  </sequence>
</complexType>
<element name="GenericAttributeSet" substitutionGroup="core:AbstractGenericAttribute" type="gen:GenericAttributeSetType">
  <annotation>
    <documentation>A GenericAttributeSet is a named collection of generic attributes.</documentation>
  </annotation>
</element>
<complexType name="GenericAttributeSetType">
  <complexContent>
    <extension base="core:AbstractGenericAttributeType">
      <sequence>
        <element name="name" type="string">
          <annotation>
            <documentation>Specifies the name of the GenericAttributeSet.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="codeSpace" type="anyURI">
          <annotation>
            <documentation>Specifies the codeSpace of the GenericAttributeSet.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<documentation>Associates the GenericAttributeSet with an authority that maintains the collection of generic attributes.</documentation>

<element maxOccurs="unbounded" name="genericAttribute" type="core:AbstractGenericAttributePropertyType">
  <annotation>
    <documentation>Relates to the generic attributes that are part of the GenericAttributeSet.</documentation>
  </annotation>
</element>

<complexType name="GenericAttributeSetPropertyType">
  <sequence>
    <element ref="gen:GenericAttributeSet"/>
  </sequence>
</complexType>

<element name="GenericLogicalSpace" substitutionGroup="core:AbstractLogicalSpace" type="gen:GenericLogicalSpaceType">
  <annotation>
    <documentation>A GenericLogicalSpace is a space that is not represented by any explicitly modelled AbstractLogicalSpace subclass within CityGML.</documentation>
  </annotation>
</element>

<complexType name="GenericLogicalSpaceType">
  <complexContent>
    <extension base="core:AbstractLogicalSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the GenericLogicalSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the GenericLogicalSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the GenericLogicalSpace.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfGenericLogicalSpace" type="gen:ADEOfGenericLogicalSpacePropertyType">
  <annotation>
    <documentation>Augments the GenericLogicalSpace with properties defined in an ADE.</documentation>
  </annotation>
</element>

<complexType name="GenericLogicalSpacePropertyType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence minOccurs="0">
        <element ref="gen:GenericLogicalSpace"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element name="GenericOccupiedSpace" substitutionGroup="core:AbstractOccupiedSpace" type="gen:GenericOccupiedSpaceType">
  <annotation>
    <documentation>A GenericOccupiedSpace is a space that is not represented by any explicitly modelled AbstractOccupiedSpace subclass within CityGML.</documentation>
  </annotation>
</element>

<complexType name="GenericOccupiedSpaceType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the GenericOccupiedSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the GenericOccupiedSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the GenericOccupiedSpace.</documentation>
          </annotation>
        </element>
      </sequence>
      <element maxOccurs="unbounded" minOccurs="0" name="adeOfGenericOccupiedSpace" type="gen:ADEOfGenericOccupiedSpacePropertyType"/>
<annotation>
    <documentation>Augments the GenericOccupiedSpace with properties defined in an ADE.</documentation>
</annotation>

<complexType name="GenericOccupiedSpacePropertyType">
    <complexContent>
        <extension base="gen:GenericOccupiedSpace"/>
    </complexContent>
</complexType>

<complexType name="GenericThematicSurfaceType">
    <complexContent>
        <extension base="core:AbstractThematicSurfaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the GenericThematicSurface.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the GenericThematicSurface.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the GenericThematicSurface.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfGenericThematicSurface" type="gen:ADEOfGenericThematicSurfacePropertyType">
                    <annotation>
                        <documentation></documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<element name="GenericThematicSurface" substitutionGroup="core:AbstractThematicSurface" type="gen:GenericThematicSurfaceType">
    <annotation>
        <documentation>A GenericThematicSurface is a surface that is not represented by any explicitly modelled AbstractThematicSurface subclass within CityGML.</documentation>
    </annotation>
</element>
Augments the GenericThematicSurface with properties defined in an ADE.

A GenericUnoccupiedSpace is a space that is not represented by any explicitly modelled AbstractUnoccupiedSpace subclass within CityGML.

A generic type of the Unoccupied Space represents a category of spaces that are not explicitly modelled as AbstractUnoccupiedSpace subclasses within CityGML.
<annotation>
<documentation>Augments the GenericUnoccupiedSpace with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</complexContent>
</complexType>
<complexType name="GenericUnoccupiedSpacePropertyType">
<sequence minOccurs="0">
<element ref="gen:GenericUnoccupiedSpace"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="IntAttribute" substitutionGroup="core:AbstractGenericAttribute" type="gen:IntAttributeType">
<annotation>
<documentation>IntAttribute is a data type used to define generic attributes of type "Integer".</documentation>
</annotation>
</element>
<complexType name="IntAttributeType">
<complexContent>
<extension base="core:AbstractGenericAttributeType">
<sequence>
<element name="name" type="string">
<annotation>
<documentation>Specifies the name of the IntAttribute.</documentation>
</annotation>
</element>
<element name="value" type="integer">
<annotation>
<documentation>Specifies the "Integer" value.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="IntAttributePropertyType">
<sequence>
<element ref="gen:IntAttribute"/>
</sequence>
</complexType>
<element name="MeasureAttribute" substitutionGroup="core:AbstractGenericAttribute" type="gen:MeasureAttributeType">
<annotation>
<documentation>MeasureAttribute is a data type used to define generic attributes of type "Measure".</documentation>
</annotation>
</element>
<complexType name="MeasureAttributeType">
    <complexContent>
        <extension base="core:AbstractGenericAttributeType">
            <sequence>
                <element name="name" type="string">
                    <annotation>
                        <documentation>Specifies the name of the MeasureAttribute.</documentation>
                    </annotation>
                </element>
                <element name="value" type="gml:MeasureType">
                    <annotation>
                        <documentation>Specifies the value of the MeasureAttribute. The value is of type "Measure", which can additionally provide the units of measure. [cf. ISO 19103]</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="MeasureAttributePropertyType">
    <sequence>
        <element ref="gen:MeasureAttribute"/>
    </sequence>
</complexType>
<element name="StringAttribute" substitutionGroup="core:AbstractGenericAttribute" type="gen:StringAttributeType">
    <annotation>
        <documentation>StringAttribute is a data type used to define generic attributes of type "String".</documentation>
    </annotation>
</element>
<complexType name="StringAttributeType">
    <complexContent>
        <extension base="core:AbstractGenericAttributeType">
            <sequence>
                <element name="name" type="string">
                    <annotation>
                        <documentation>Specifies the name of the StringAttribute.</documentation>
                    </annotation>
                </element>
                <element name="value" type="string">
                    <annotation>
                        <documentation>Specifies the "String" value.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
C.10. Land Use

The CityGML Land Use module is defined in the XML Schema Definition file landUse.xsd (Listing 37). The target namespace http://www.opengis.net/citygml/landuse/3.0 is associated with this module.

Listing 37. landUse.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:core="http://www.opengis.net/citygml/3.0"
```
The LandUse module supports representation of areas of the earth's surface dedicated to a specific land use.

ADEOfLandUse acts as a hook to define properties within an ADE that are to be added to a LandUse.

A LandUse object is an area of the earth's surface dedicated to a specific land use or having a specific land cover with or without vegetation, such as sand, rock, mud flats, forest, grasslands, or wetlands.
C.11. Point Cloud

The CityGML Point Cloud module is defined in the XML Schema Definition file `pointCloud.xsd` (Listing 38). The target namespace `http://www.opengis.net/citygml/pointcloud/3.0` is associated with this module.

Listing 38. pointCloud.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?><schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:core="http://www.opengis.net/citygml/3.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:pcl="http://www.opengis.net/citygml/pointcloud/3.0"
elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/citygml/pointcloud/3.0" version="3.0.0">
  <annotation>
    <documentation>The PointCloud module supports representation of CityGML features by a collection of points.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net/-->
  <element abstract="true" name="ADEOfPointCloud" substitutionGroup="gml:AbstractObject" type="pcl:ADEOfPointCloudType">
```
ADEOfPointCloud acts as a hook to define properties within an ADE that are to be added to a PointCloud.

A PointCloud is an unordered collection of points that is a sampling of the geometry of a space or space boundary.

PointCloud is an unordered collection of points that is a sampling of the geometry of a space or space boundary.

- mimeType: Specifies the MIME type of the external point cloud file.
- pointFile: Specifies the URI that points to the external point cloud file.
- pointFileSrsName: Indicates the coordinate reference system used by the external point cloud file.
- points: Relates to the 3D MultiPoint geometry of the PointCloud stored inline with the city model.
- adeOfPointCloud: Additional properties specific to the PointCloud.
C.12. Relief

The CityGML Relief module is defined in the XML Schema Definition file relief.xsd (Listing 39). The target namespace http://www.opengis.net/citygml/relief/3.0 is associated with this module.

Listing 39. relief.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:core="http://www.opengis.net/citygml/3.0"
xmlns:dem="http://www.opengis.net/citygml/relief/3.0"
xmlns:gml="http://www.opengis.net/gml/3.2" elementFormDefault="qualified"
targetNamespace="http://www.opengis.net/citygml/relief/3.0" version="3.0.0">
  <annotation>
    <documentation>The Relief module supports representation of the terrain. CityGML supports terrain representations at different levels of detail, reflecting different accuracies or resolutions. Terrain may be specified as a regular raster or grid, as a TIN, by break lines, and/or by mass points.</documentation>
  </annotation>
  <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
  <import namespace="http://www.opengis.net/gml/3.2"
schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net/-->
  <element abstract="true" name="ADEOfAbstractReliefComponent"
substitutionGroup="gml:AbstractObject" type="dem:ADEOfAbstractReliefComponentType">
    <annotation>
      <documentation>ADEOfAbstractReliefComponent acts as a hook to define properties within an ADE that are to be added to AbstractReliefComponent.</documentation>
    </annotation>
    <complexType abstract="true" name="ADEOfAbstractReliefComponentType">
      <sequence/>
    </complexType>
  </element>
</schema>
```
<element abstract="true" name="ADEOfAbstractReliefComponent"/>
</element>
</complexType>

<complexType>
<element abstract="true" name="ADEOfBreaklineRelief" substitutionGroup="gml:AbstractObject" type="dem:ADEOfBreaklineReliefType">
<annotation>
<documentation>ADEOfBreaklineRelief acts as a hook to define properties within an ADE that are to be added to a BreaklineRelief.</documentation>
</annotation>
</element>
</complexType>

<complexType abstract="true" name="ADEOfBreaklineReliefType">
<sequence/>
</complexType>

<complexType name="ADEOfBreaklineReliefPropertyType">
<sequence>
<element ref="dem:ADEOfBreaklineRelief"/>
</sequence>
</complexType>

<element abstract="true" name="ADEOfMassPointRelief" substitutionGroup="gml:AbstractObject" type="dem:ADEOfMassPointReliefType">
<annotation>
<documentation>ADEOfMassPointRelief acts as a hook to define properties within an ADE that are to be added to a MassPointRelief.</documentation>
</annotation>
</element>
</complexType>

<complexType abstract="true" name="ADEOfMassPointReliefType">
<sequence/>
</complexType>

<complexType name="ADEOfMassPointReliefPropertyType">
<sequence>
<element ref="dem:ADEOfMassPointRelief"/>
</sequence>
</complexType>

<element abstract="true" name="ADEOfRasterRelief" substitutionGroup="gml:AbstractObject" type="dem:ADEOfRasterReliefType">
<annotation>
<documentation>ADEOfRasterRelief acts as a hook to define properties within an ADE that are to be added to a RasterRelief.</documentation>
</annotation>
</element>
</complexType>

<complexType abstract="true" name="ADEOfRasterReliefType">
<sequence/>
</complexType>

<complexType name="ADEOfRasterReliefPropertyType">
<sequence>
<element ref="dem:ADEOfRasterRelief"/>
</sequence>
</complexType>

<element abstract="true" name="ADEOfReliefFeature" substitutionGroup="gml:AbstractObject" type="dem:ADEOfReliefFeatureType"/>
<documentation>ADEOfReliefFeature acts as a hook to define properties within an ADE that are to be added to a ReliefFeature.</documentation>
</annotation>
</element>
<complexType abstract="true" name="ADEOfReliefFeatureType">
  <sequence/>
</complexType>
<complexType name="ADEOfReliefFeaturePropertyType">
  <sequence>
    <element ref="dem:ADEOfReliefFeature"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfTINRelief" substitutionGroup="gml:AbstractObject" type="dem:ADEOfTINReliefType">
  <annotation>
    <documentation>ADEOfTINRelief acts as a hook to define properties within an ADE that are to be added to a TINRelief.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfTINReliefType">
  <sequence/>
</complexType>
<complexType name="ADEOfTINReliefPropertyType">
  <sequence>
    <element ref="dem:ADEOfTINRelief"/>
  </sequence>
</complexType>
<element abstract="true" name="AbstractReliefComponent" substitutionGroup="core:AbstractSpaceBoundary" type="dem:AbstractReliefComponentType">
  <annotation>
    <documentation>An AbstractReliefComponent represents an element of the terrain surface - either a TIN, a raster or grid, mass points or break lines.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractReliefComponentType">
  <complexContent>
    <extension base="core:AbstractSpaceBoundaryType">
      <sequence>
        <element name="lod" type="core:IntegerBetween0and3Type">
          <annotation>
            <documentation>Indicates the Level of Detail of the terrain component.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="extent" type="gml:SurfacePropertyType">
          <annotation>
            <documentation>Indicates the geometrical extent of the terrain component. The geometrical extent is provided as a 2D Surface geometry.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractReliefComponent" type="dem:ADEOfAbstractReliefComponentPropertyType">
<annotation>
<documentation>Augments AbstractReliefComponent with properties defined in an ADE.</documentation>
</annotation>
</element>
<complexType name="AbstractReliefComponentPropertyType">
<sequence minOccurs="0">
<element ref="dem:AbstractReliefComponent"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="BreaklineRelief" substitutionGroup="dem:AbstractReliefComponent" type="dem:BreaklineReliefType">
<annotation>
<documentation>A BreaklineRelief represents a terrain component with 3D lines. These lines denote break lines or ridge/valley lines.</documentation>
</annotation>
</element>
<complexType name="BreaklineReliefType">
<complexContent>
<extension base="dem:AbstractReliefComponentType">
<sequence>
<element minOccurs="0" name="ridgeOrValleyLines" type="gml:MultiCurvePropertyType">
<annotation>
<documentation>Relates to the 3D MultiCurve geometry of the MassPointRelief. This association role is used to represent ridge or valley lines.</documentation>
</annotation>
</element>
<element minOccurs="0" name="breaklines" type="gml:MultiCurvePropertyType">
<annotation>
<documentation>Relates to the 3D MultiCurve geometry of the MassPointRelief. This association role is used to represent break lines.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBreaklineRelief" type="dem:ADEOfBreaklineReliefPropertyType">
<annotation>
<documentation>Augments the BreaklineRelief with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="BreaklineReliefPropertyType">
  <sequence minOccurs="0">
    <element ref="dem:BreaklineRelief"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="MassPointRelief" substitutionGroup="dem:AbstractReliefComponent" type="dem:MassPointReliefType">
  <annotation>
    <documentation>A MassPointRelief represents a terrain component as a collection of 3D points.</documentation>
  </annotation>
</element>
<complexType name="MassPointReliefType">
  <complexContent>
    <extension base="dem:AbstractReliefComponentType">
      <sequence>
        <element minOccurs="0" name="reliefPoints" type="gml:MultiPointPropertyType">
          <annotation>
            <documentation>Relates to the 3D MultiPoint geometry of the MassPointRelief.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="pointCloud" type="core:AbstractPointCloudPropertyType">
          <annotation>
            <documentation>Relates to the 3D PointCloud of the MassPointRelief.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfMassPointRelief" type="dem:ADEOfMassPointReliefPropertyType">
          <annotation>
            <documentation>Augments the MassPointRelief with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="MassPointReliefPropertyType">
  <sequence minOccurs="0">
  </sequence>
</complexType>
<element name="RasterRelief" substitutionGroup="dem:AbstractReliefComponent" type="dem:RasterReliefType">
  <annotation>
    <documentation>A RasterRelief represents a terrain component as a regular raster or grid.</documentation>
  </annotation>
</element>

<complexType name="RasterReliefType">
  <complexContent>
    <extension base="dem:AbstractReliefComponentType">
      <sequence>
        <element name="grid">
          <annotation>
            <documentation>Relates to the DiscreteGridPointCoverage of the RasterRelief.</documentation>
          </annotation>
          <complexType>
            <sequence minOccurs="0">
              <element ref="gml:RectifiedGridCoverage"/>
            </sequence>
            <attributeGroup ref="gml:AssociationAttributeGroup"/>
            <attributeGroup ref="gml:OwnershipAttributeGroup"/>
          </complexType>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfRasterRelief" type="dem:ADEOfRasterReliefPropertyType">
          <annotation>
            <documentation>Augments the RasterRelief with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="ReliefFeature" substitutionGroup="core:AbstractSpaceBoundary" type="dem:ReliefFeatureType">
  <annotation>
    <documentation>A ReliefFeature is a collection of terrain components
    </documentation>
  </annotation>
</complexType>
representing the Earth's surface, also known as the Digital Terrain Model.</annotation></element></complexType>
<complexType name="ReliefFeatureType">
  <complexContent>
    <extension base="core:AbstractSpaceBoundaryType">
      <sequence>
        <element name="lod" type="core:IntegerBetween0and3Type">
          <annotation>
            <documentation>Indicates the Level of Detail of the ReliefFeature.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" name="reliefComponent">
          <annotation>
            <documentation>Relates to the terrain components that are part of the ReliefFeature.</documentation>
          </annotation>
          <complexType>
            <complexContent>
              <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                  <element ref="dem:AbstractReliefComponent"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfReliefFeature" type="dem:ADEOfReliefFeaturePropertyType">
          <annotation>
            <documentation>Augments the ReliefFeature with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="ReliefFeaturePropertyType">
  <sequence minOccurs="0">
    <element ref="dem:ReliefFeature"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TINRelief" substitutionGroup="dem:AbstractReliefComponent" type="dem:TINReliefType">
  <annotation>

278
A TINRelief represents a terrain component as a triangulated irregular network.

```xml
<complexType name="TINReliefType">
  <complexContent>
    <extension base="dem:AbstractReliefComponentType">
      <sequence>
        <element name="tin">
          <annotation>
            <documentation>Relates to the triangulated surface of the TINRelief.</documentation>
          </annotation>
          <complexType>
            <sequence minOccurs="0">
              <element ref="gml:TriangulatedSurface"/>
            </sequence>
            <attributeGroup ref="gml:AssociationAttributeGroup"/>
            <attributeGroup ref="gml:OwnershipAttributeGroup"/>
          </complexType>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfTINRelief" type="dem:ADEOfTINReliefPropertyType">
          <annotation>
            <documentation>Augments the TINRelief with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

C.13. Transportation

The CityGML Transportation module is defined in the XML Schema Definition file transportation.xsd (Listing 40). The target namespace http://www.opengis.net/citygml/transportation/3.0 is associated with this module.

Listing 40. transportation.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
```
The Transportation module supports representation of the transportation infrastructure. Transportation features include roads, tracks, waterways, railways, and squares. Transportation features may be represented as a network and/or as a collection of spaces or surface elements embedded in a three-dimensional space.

ADEOfAbstractTransportationSpace acts as a hook to define properties within an ADE that are to be added to AbstractTransportationSpace.

ADEOfAuxiliaryTrafficArea acts as a hook to define properties within an ADE that are to be added to an AuxiliaryTrafficArea.

ADEOfAuxiliaryTrafficSpace acts as a hook to define properties within an ADE that are to be added to an AuxiliaryTrafficArea.
<documentation>ADEOfAuxiliaryTrafficSpace acts as a hook to define properties within an ADE that are to be added to an AuxiliaryTrafficSpace.</documentation>
</annotation>

<complexType abstract="true" name="ADEOfAuxiliaryTrafficSpaceType">
  <sequence/>
</complexType>

<complexType name="ADEOfAuxiliaryTrafficSpacePropertyType">
  <sequence>
    <element ref="tran:ADEOfAuxiliaryTrafficSpace"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfClearanceSpace" substitutionGroup="gml:AbstractObject" type="tran:ADEOfClearanceSpaceType">
  <annotation>
    <documentation>ADEOfClearanceSpace acts as a hook to define properties within an ADE that are to be added to a ClearanceSpace.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfClearanceSpaceType">
  <sequence/>
</complexType>

<complexType name="ADEOfClearanceSpacePropertyType">
  <sequence>
    <element ref="tran:ADEOfClearanceSpace"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfHole" substitutionGroup="gml:AbstractObject" type="tran:ADEOfHoleType">
  <annotation>
    <documentation>ADEOfHole acts as a hook to define properties within an ADE that are to be added to a Hole.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfHoleType">
  <sequence/>
</complexType>

<complexType name="ADEOfHolePropertyType">
  <sequence>
    <element ref="tran:ADEOfHole"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfHoleSurface" substitutionGroup="gml:AbstractObject" type="tran:ADEOfHoleSurfaceType">
  <annotation>
    <documentation>ADEOfHoleSurface acts as a hook to define properties within an ADE that are to be added to a HoleSurface.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfHoleSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfHoleSurfacePropertyType">
  <sequence>
    <element ref="tran:ADEOfHoleSurface"/>
  </sequence>
</complexType>

<complexType name="ADEOfIntersectionPropertyType">
  <sequence>
    <element ref="tran:ADEOfIntersection"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfIntersection" substitutionGroup="gml:AbstractObject" type="tran:ADEOfIntersectionType">
  <annotation>
    <documentation>ADEOfIntersection acts as a hook to define properties within an ADE that are to be added to an Intersection.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfIntersectionType">
  <sequence/>
</complexType>

<complexType name="ADEOfIntersectionPropertyType">
  <sequence>
    <element ref="tran:ADEOfIntersection"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfMarking" substitutionGroup="gml:AbstractObject" type="tran:ADEOfMarkingType">
  <annotation>
    <documentation>ADEOfMarking acts as a hook to define properties within an ADE that are to be added to a Marking.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfMarkingType">
  <sequence/>
</complexType>

<complexType name="ADEOfMarkingPropertyType">
  <sequence>
    <element ref="tran:ADEOfMarking"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfRailway" substitutionGroup="gml:AbstractObject" type="tran:ADEOfRailwayType">
  <annotation>
    <documentation>ADEOfRailway acts as a hook to define properties within an ADE that are to be added to a Railway.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfRailwayType">
  <sequence/>
</complexType>

<complexType name="ADEOfRailwayPropertyType">
  <sequence>
    <element ref="tran:ADEOfRailway"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfRoad" substitutionGroup="gml:AbstractObject" type="tran:ADEOfRoadType">
  <annotation>
    <documentation>ADEOfRoad acts as a hook to define properties within an ADE that are to be added to a Road.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfRoadType">
  <sequence/>
</complexType>
<complexType name="ADEOfRoadPropertyType">
  <sequence>
    <element ref="tran:ADEOfRoad"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfSection" substitutionGroup="gml:AbstractObject" type="tran:ADEOfSectionType">
  <annotation>
    <documentation>ADEOfSection acts as a hook to define properties within an ADE that are to be added to a Section.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfSectionType">
  <sequence/>
</complexType>
<complexType name="ADEOfSectionPropertyType">
  <sequence>
    <element ref="tran:ADEOfSection"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfSquare" substitutionGroup="gml:AbstractObject" type="tran:ADEOfSquareType">
  <annotation>
    <documentation>ADEOfSquare acts as a hook to define properties within an ADE that are to be added to a Square.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfSquareType">
  <sequence/>
</complexType>
<complexType name="ADEOfSquarePropertyType">
  <sequence>
    <element ref="tran:ADEOfSquare"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfTrack" substitutionGroup="gml:AbstractObject" type="tran:ADEOfTrackType">
  <annotation>
    <documentation>ADEOfTrack acts as a hook to define properties within an ADE that are to be added to a Track.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfTrackType">
    <sequence/>
</complexType>

<complexType name="ADEOfTrackPropertyType">
    <element ref="tran:ADEOfTrack"/>
</complexType>

<element abstract="true" name="ADEOfTrafficArea" substitutionGroup="gml:AbstractObject" type="tran:ADEOfTrafficAreaType">
    <annotation>
        <documentation>ADEOfTrafficArea acts as a hook to define properties within an ADE that are to be added to a TrafficArea.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfTrafficAreaType">
    <sequence/>
</complexType>

<complexType name="ADEOfTrafficAreaPropertyType">
    <element ref="tran:ADEOfTrafficArea"/>
</complexType>

<element abstract="true" name="ADEOfTrafficSpace" substitutionGroup="gml:AbstractObject" type="tran:ADEOfTrafficSpaceType">
    <annotation>
        <documentation>ADEOfTrafficSpace acts as a hook to define properties within an ADE that are to be added to a TrafficSpace.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfTrafficSpaceType">
    <sequence/>
</complexType>

<complexType name="ADEOfTrafficSpacePropertyType">
    <element ref="tran:ADEOfTrafficSpace"/>
</complexType>

<element abstract="true" name="ADEOfWaterway" substitutionGroup="gml:AbstractObject" type="tran:ADEOfWaterwayType">
    <annotation>
        <documentation>ADEOfWaterway acts as a hook to define properties within an ADE that are to be added to a Waterway.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfWaterwayType">
    <sequence/>
</complexType>

<complexType name="ADEOfWaterwayPropertyType">
    <sequence/>
</complexType>
<element ref="tran:ADEOfWaterway"/>
</sequence>
<complexType>
<element abstract="true" name="AbstractTransportationSpace"
substitutionGroup="core:AbstractUnoccupiedSpace" type="tran:AbstractTransportationSpaceType">
<annotation>
<documentation>AbstractTransportationSpace is the abstract superclass of transportation objects such as Roads, Tracks, Railways, Waterways or Squares.</documentation>
</annotation>
</element>
<complexType abstract="true" name="AbstractTransportationSpaceType">
<complexContent>
<extension base="core:AbstractUnoccupiedSpaceType">
<sequence>
<element minOccurs="0" name="trafficDirection" type="tran:TrafficDirectionValueType">
<annotation>
<documentation>Indicates the direction of traffic flow relative to the corresponding linear geometry representation.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="occupancy" type="core:OccupancyPropertyType">
<annotation>
<documentation>Provides information on the residency of persons, vehicles, or other moving features in the transportation space.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="trafficSpace">
<annotation>
<documentation>Relates to the traffic spaces that are part of the transportation space.</documentation>
</annotation>
<complexType>
<complexContent>
<extension base="gml:AbstractFeatureMemberType">
<sequence minOccurs="0">
<element ref="tran:TrafficSpace"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>
</complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="auxiliaryTrafficSpace">
<annotation>
<documentation>Relates to the auxiliary traffic spaces that are part of the transportation space.</documentation>
</annotation>
</element>
</extension>
</complexContent>
</complexType>
<complexType>
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="tran:AuxiliaryTrafficSpace"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element maxOccurs="unbounded" minOccurs="0" name="hole">
  <annotation>
    <documentation>Relates to the holes that are part of the transportation space.</documentation>
  </annotation>
</element>

<complexType>
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="tran:Hole"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element maxOccurs="unbounded" minOccurs="0" name="marking">
  <annotation>
    <documentation>Relates to the markings that are part of the transportation space.</documentation>
  </annotation>
</element>

<complexType>
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="tran:Marking"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractTransportationSpace" type=" tran:ADEOfAbstractTransportationSpacePropertyType">
  <annotation>
    <documentation>Augments AbstractTransportationSpace with properties defined in an ADE.</documentation>
  </annotation>
</element>
<complexType name="AbstractTransportationSpacePropertyType">
    <sequence minOccurs="0">
        <element ref="tran:AbstractTransportationSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="AuxiliaryTrafficArea" substitutionGroup="core:AbstractThematicSurface" type="tran:AuxiliaryTrafficAreaType">
    <annotation>
        <documentation>An AuxiliaryTrafficArea is the ground surface of an AuxiliaryTrafficSpace.</documentation>
    </annotation>
</element>
<complexType name="AuxiliaryTrafficAreaType">
    <complexContent>
        <extension base="core:AbstractThematicSurfaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the AuxiliaryTrafficArea.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the AuxiliaryTrafficArea.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the AuxiliaryTrafficArea.</documentation>
                    </annotation>
                </element>
                <element minOccurs="0" name="surfaceMaterial" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the type of pavement of the AuxiliaryTrafficArea.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfAuxiliaryTrafficArea" type="tran:ADEOfAuxiliaryTrafficAreaPropertyType">
                    <annotation>
                        <documentation></documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
Augments the AuxiliaryTrafficArea with properties defined in an ADE.

```xml
<complexType name="AuxiliaryTrafficAreaPropertyType">
  <sequence minOccurs="0">
    <element ref="tran:AuxiliaryTrafficArea"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

An AuxiliaryTrafficSpace is a space within the transportation space not intended for traffic purposes.

```xml
<complexType name="AuxiliaryTrafficSpaceType">
  <complexContent>
    <extension base="core:AbstractUnoccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the AuxiliaryTrafficSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the AuxiliaryTrafficSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the AuxiliaryTrafficSpace.</documentation>
          </annotation>
        </element>
        <element name="granularity" type="tran:GranularityValueType">
          <annotation>
            <documentation>Defines whether auxiliary traffic spaces are represented by individual ways or by individual lanes, depending on the desired level of spatial precision.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```
and semantic decomposition.</documentation>
</element>
</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfAuxiliaryTrafficSpace" type="tran:ADEOfAuxiliaryTrafficSpacePropertyType">
<annotation>
<documentation>Augments the AuxiliaryTrafficSpace with properties defined in an ADE.</documentation>
</annotation>
</element>
</extension>
</complexContent>
</complexType>
<complexType name="AuxiliaryTrafficSpacePropertyType">
<sequence minOccurs="0">
<element ref="tran:AuxiliaryTrafficSpace"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="ClearanceSpace" substitutionGroup="core:AbstractUnoccupiedSpace" type="tran:ClearanceSpaceType">
<annotation>
<documentation>A ClearanceSpace represents the actual free space above a TrafficArea within which a mobile object can move without contacting an obstruction.</documentation>
</annotation>
</element>
<complexType name="ClearanceSpaceType">
<complexContent>
<extension base="core:AbstractUnoccupiedSpaceType">
<sequence>
<element maxOccurs="unbounded" minOccurs="0" name="class" type="gml:CodeType">
<annotation>
<documentation>Indicates the specific type of the ClearanceSpace.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfClearanceSpace" type="tran:ADEOfClearanceSpacePropertyType">
<annotation>
<documentation>Augments the ClearanceSpace with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="ClearanceSpacePropertyType">
    <sequence minOccurs="0">
        <element ref="tran:ClearanceSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<simpleType name="GranularityValueType">
    <annotation>
        <documentation>GranularityValue enumerates the different levels of granularity in which transportation objects are represented.</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="lane">
            <annotation>
                <documentation>Indicates that the individual lanes of the transportation object are represented.</documentation>
            </annotation>
        </enumeration>
        <enumeration value="way">
            <annotation>
                <documentation>Indicates that the individual (carriage)ways of the transportation object are represented.</documentation>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>

<element name="Hole" substitutionGroup="core:AbstractUnoccupiedSpace" type="tran:HoleType">
    <annotation>
        <documentation>A Hole is an opening in the surface of a Road, Track or Square such as road damages, manholes or drains. Holes can span multiple transportation objects.</documentation>
    </annotation>
</element>

<complexType name="HoleType">
    <complexContent>
        <extension base="core:AbstractUnoccupiedSpaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the Hole.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfHole" type="tran:ADEOfHolePropertyType">
                    <annotation>
                        <documentation>Augments the Hole with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="HolePropertyType">
  <sequence minOccurs="0">
    <element ref="tran:Hole"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="HoleSurface" substitutionGroup="core:AbstractThematicSurface" type="tran:HoleSurfaceType">
  <annotation>
    <documentation>A HoleSurface is a representation of the ground surface of a hole.</documentation>
  </annotation>
</element>

<complexType name="HoleSurfaceType">
  <complexContent>
    <extension base="core:AbstractThematicSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfHoleSurface" type="tran:ADEOfHoleSurfacePropertyType">
          <annotation>
            <documentation>Augments the HoleSurface with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="HoleSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="tran:HoleSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="Intersection" substitutionGroup="tran:AbstractTransportationSpace" type="tran:IntersectionType">
  <annotation>
    <documentation>An Intersection is a transportation space that is a shared segment of multiple Road, Track, Railway, or Waterway objects (e.g. a crossing of two roads or a level crossing of a road and a railway).</documentation>
  </annotation>
</element>

<complexType name="IntersectionType">
  <complexContent>
    <extension base="tran:AbstractTransportationSpaceType">
      ...
    </extension>
  </complexContent>
</complexType>
<sequence>
  <element minOccurs="0" name="class" type="gml:CodeType">
    <annotation>
      <documentation>Indicates the specific type of the Intersection.</documentation>
    </annotation>
  </element>
  <element maxOccurs="unbounded" minOccurs="0" name="adeOfIntersection" type="tran:ADEOfIntersectionPropertyType">
    <annotation>
      <documentation>Augments the Intersection with properties defined in an ADE.</documentation>
    </annotation>
  </element>
</sequence>

<complexType name="IntersectionPropertyType">
  <sequence minOccurs="0">
    <element ref="tran:Intersection"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="Marking" substitutionGroup="core:AbstractThematicSurface" type="tran:MarkingType">
  <annotation>
    <documentation>A Marking is a visible pattern on a transportation area relevant to the structuring or restriction of traffic. Examples are road markings and markings related to railway or waterway traffic.</documentation>
  </annotation>
</element>

<complexType name="MarkingType">
  <complexContent>
    <extension base="core:AbstractThematicSurfaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the Marking.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfMarking" type="tran:ADEOfMarkingPropertyType">
          <annotation>
            <documentation>Augments the Marking with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="MarkingPropertyType">
  <sequence minOccurs="0">
    <element ref="tran:Marking"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Railway" substitutionGroup="tran:AbstractTransportationSpace" type="tran:RailwayType">
  <annotation>
    <documentation>A Railway is a transportation space used by wheeled vehicles on rails.</documentation>
  </annotation>
</element>
<complexType name="RailwayType">
  <complexContent>
    <extension base="tran:AbstractTransportationSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the Railway.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the Railway.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the Railway.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="section">
          <complexType>
            <complexContent>
              <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                  <element ref="tran:Section"/>
                </sequence>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="intersection">
  <annotation>
    <documentation>Relates to the intersections that are part of the Railway.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="tran:Intersection"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfRailway">
  <annotation>
    <documentation>Augments the Railway with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="RailwayPropertyType">
  <sequence minOccurs="0">
    <element ref="tran:Railway"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Road" substitutionGroup="tran:AbstractTransportationSpace" type="tran:RoadType">
  <annotation>
    <documentation>A Road is a transportation space used by vehicles, bicycles and/or pedestrians.</documentation>
  </annotation>
</element>
<complexType name="RoadType">
  <complexContent>
    <extension base="tran:AbstractTransportationSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          294
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
Indicates the specific type of the Road.

Specifies the intended purposes of the Road.

Specifies the actual uses of the Road.

Relates to the sections that are part of the Road.

Relates to the intersections that are part of the Road.
Augments the Road with properties defined in an ADE.

A Section is a transportation space that is a segment of a Road, Railway, Track, or Waterway.

Indicates the specific type of the Section.

Augments the Section with properties defined in an ADE.
<element name="Square" substitutionGroup="tran:AbstractTransportationSpace" type="tran:SquareType">
    <annotation>
        <documentation>A Square is a transportation space for unrestricted movement for vehicles, bicycles and/or pedestrians. This includes plazas as well as large sealed surfaces such as parking lots.</documentation>
    </annotation>
</element>
<complexType name="SquareType">
    <complexContent>
        <extension base="tran:AbstractTransportationSpaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the Square.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the Square.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the Square.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfSquare" type="tran:ADEOfSquarePropertyType">
                    <annotation>
                        <documentation>Augments the Square with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="SquarePropertyType">
    <sequence minOccurs="0">
        <element ref="tran:Square"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<complexType name="Track" substitutionGroup="tran:AbstractTransportationSpace" type="tran:TrackType">
A Track is a small path mainly used by pedestrians. Tracks can be segmented into Sections and Intersections.

<complexType name="TrackType">
  <complexContent>
    <extension base="tran:AbstractTransportationSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the Track.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the Track.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the Track.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="section">
          <annotation>
            <documentation>Relates to the sections that are part of the Track.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="intersection">
          <annotation>
            <documentation>Relates to the intersections that are part of the Track.</documentation>
          </annotation>
        </element>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<sequence minOccurs="0">
  <element ref="tran:Intersection"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfTrack" type="tran:ADEOfTrackPropertyType">
  <annotation>
    <documentation>Augments the Track with properties defined in an ADE.</documentation>
  </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TrackPropertyType">
  <sequence minOccurs="0">
    <element ref="tran:Track"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TrafficArea" substitutionGroup="core:AbstractThematicSurface" type="tran:TrafficAreaType">
  <annotation>
    <documentation>A TrafficArea is the ground surface of a TrafficSpace. Traffic areas are the surfaces upon which traffic actually takes place.</documentation>
  </annotation>
</element>
<complexType name="TrafficAreaType">
  <complexContent>
    <extension base="core:AbstractThematicSurfaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the TrafficArea.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the TrafficArea.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage"
<complexType name="TrafficAreaPropertyType">
  <sequence minOccurs="0">
    <element ref="tran:TrafficArea" />
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup" />
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

<complexType name="TrafficDirectionValueType">
  <annotation>
    <documentation>TrafficDirectionValue enumerates the allowed directions of travel of a mobile object.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="forwards">
      <annotation>
        <documentation>Indicates that traffic flows in the direction of the corresponding linear geometry.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="backwards">
      <annotation>
        <documentation>Indicates that traffic flows in the opposite direction of the corresponding linear geometry.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="both">
      <annotation>
        <documentation>Indicates that traffic flows in both
<element name="TrafficSpace" substitutionGroup="core:AbstractUnoccupiedSpace" type="tran:TrafficSpaceType">
  <annotation>
    <documentation>A TrafficSpace is a space in which traffic takes place. Traffic includes the movement of entities such as trains, vehicles, pedestrians, ships, or other transportation types.</documentation>
  </annotation>
</element>
<complexType name="TrafficSpaceType">
  <complexContent>
    <extension base="core:AbstractUnoccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the TrafficSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the TrafficSpace.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the TrafficSpace.</documentation>
          </annotation>
        </element>
        <element name="granularity" type="tran:GranularityValueType">
          <annotation>
            <documentation>Defines whether traffic spaces are represented by individual ways or by individual lanes, depending on the desired level of spatial and semantic decomposition.</documentation>
          </annotation>
        </element>
        <element minOccurs="0" name="trafficDirection" type="tran:TrafficDirectionValueType">
          <annotation>
            <documentation>Indicates the direction of traffic flow relative to the corresponding linear geometry representation.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="occupancy" type="core:OccupancyPropertyType">
    <annotation>
        <documentation>Provides information on the residency of persons, vehicles, or other moving features in the TrafficSpace.</documentation>
    </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="predecessor" type="gml:ReferenceType">
    <annotation>
        <documentation>Indicates the predecessor(s) of the TrafficSpace.</documentation>
    </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="successor" type="gml:ReferenceType">
    <annotation>
        <documentation>Indicates the successor(s) of the TrafficSpace.</documentation>
    </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="clearanceSpace">
    <annotation>
        <documentation>Relates to the clearance spaces that are part of the TrafficSpace.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tran:ClearanceSpace"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfTrafficSpace" type="tran:ADEOfTrafficSpacePropertyType">
    <annotation>
        <documentation>Augments the TrafficSpace with properties defined in an ADE.</documentation>
    </annotation>
</element>
<element name="Waterway" substitutionGroup="tran:AbstractTransportationSpace" type="tran:WaterwayType">
  <annotation>
    <documentation>A Waterway is a transportation space used for the movement of vessels upon or within a water body.</documentation>
  </annotation>
</element>

<complexType name="WaterwayType">
  <complexContent>
    <extension base="tran:AbstractTransportationSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the Waterway.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the Waterway.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the Waterway.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="section">
          <annotation>
            <documentation>Relates to the sections that are part of the Waterway.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
C.14. Tunnel

The CityGML Tunnel module is defined in the XML Schema Definition file tunnel.xsd (Listing 41). The target namespace http://www.opengis.net/citygml/tunnel/3.0 is associated with this module.

Listing 41. tunnel.xsd
<annotation>
  <documentation>The Tunnel module supports representation of thematic and spatial aspects of tunnels, tunnel parts, tunnel installations, and interior tunnel structures.</documentation>
</annotation>
<import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
<import namespace="http://www.opengis.net/citygml/construction/3.0" schemaLocation="./construction.xsd"/>
<import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!--XML Schema document created by ShapeChange - http://shapechange.net-->  
  <element abstract="true" name="ADEOfAbstractTunnel" substitutionGroup="gml:AbstractObject" type="tun:ADEOfAbstractTunnelType">
    <annotation>
      <documentation>ADEOfAbstractTunnel acts as a hook to define properties within an ADE that are to be added to AbstractTunnel.</documentation>
    </annotation>
    </element>
    <complexType abstract="true" name="ADEOfAbstractTunnelType">
      <sequence/>
    </complexType>
    <complexType name="ADEOfAbstractTunnelPropertyType">
      <sequence>
        <element ref="tun:ADEOfAbstractTunnel"/>
      </sequence>
    </complexType>
    <element abstract="true" name="ADEOfHollowSpace" substitutionGroup="gml:AbstractObject" type="tun:ADEOfHollowSpaceType">
      <annotation>
        <documentation>ADEOfHollowSpace acts as a hook to define properties within an ADE that are to be added to a HollowSpace.</documentation>
      </annotation>
    </element>
    <complexType abstract="true" name="ADEOfHollowSpaceType">
      <sequence/>
    </complexType>
    <complexType name="ADEOfHollowSpacePropertyType">
      <sequence>
        <element ref="tun:ADEOfHollowSpace"/>
      </sequence>
    </complexType>
    <element abstract="true" name="ADEOfTunnel" substitutionGroup="gml:AbstractObject" type="tun:ADEOfTunnelType">
      <annotation>
        <documentation>ADEOfTunnel acts as a hook to define properties within an ADE that are to be added to a Tunnel.</documentation>
      </annotation>
    </element>
    <complexType abstract="true" name="ADEOfTunnelType">
      <sequence/>
    </complexType>
    <complexType name="ADEOfTunnelPropertyType">
      <sequence>
        <element ref="tun:ADEOfTunnel"/>
      </sequence>
    </complexType>
that are to be added to a Tunnel.</documentation>

<complexType abstract="true" name="ADEOfTunnelType">
  <sequence/>
</complexType>

<complexType name="ADEOfTunnelPropertyType">
  <sequence>
    <element ref="tun:ADEOfTunnel"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfTunnelConstructiveElement" substitutionGroup="gml:AbstractObject" type="tun:ADEOfTunnelConstructiveElementType">
  <annotation>
    <documentation>ADEOfTunnelConstructiveElement acts as a hook to define properties within an ADE that are to be added to a TunnelConstructiveElement.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfTunnelConstructiveElementType">
  <sequence/>
</complexType>

<complexType name="ADEOfTunnelConstructiveElementPropertyType">
  <sequence>
    <element ref="tun:ADEOfTunnelConstructiveElement"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfTunnelFurniture" substitutionGroup="gml:AbstractObject" type="tun:ADEOfTunnelFurnitureType">
  <annotation>
    <documentation>ADEOfTunnelFurniture acts as a hook to define properties within an ADE that are to be added to a TunnelFurniture.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfTunnelFurnitureType">
  <sequence/>
</complexType>

<complexType name="ADEOfTunnelFurniturePropertyType">
  <sequence>
    <element ref="tun:ADEOfTunnelFurniture"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfTunnelInstallation" substitutionGroup="gml:AbstractObject" type="tun:ADEOfTunnelInstallationType">
  <annotation>
    <documentation>ADEOfTunnelInstallation acts as a hook to define properties within an ADE that are to be added to a TunnelInstallation.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfTunnelInstallationType">
  <sequence/>
</complexType>
<complexType name="ADEOfTunnelInstallationPropertyType">
    <sequence>
        <element ref="tun:ADEOfTunnelInstallation"/>
    </sequence>
</complexType>

<element abstract="true" name="ADEOfTunnelPart" substitutionGroup="gml:AbstractObject" type="tun:ADEOfTunnelPartType">
    <annotation>
        <documentation>ADEOfTunnelPart acts as a hook to define properties within an ADE that are to be added to a TunnelPart.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="ADEOfTunnelPartType">
    <sequence/>
</complexType>

<complexType name="ADEOfTunnelPartPropertyType">
    <sequence>
        <element ref="tun:ADEOfTunnelPart"/>
    </sequence>
</complexType>

<element abstract="true" name="AbstractTunnel" substitutionGroup="con:AbstractConstruction" type="tun:AbstractTunnelType">
    <annotation>
        <documentation>AbstractTunnel is an abstract superclass representing the common attributes and associations of the classes Tunnel and TunnelPart.</documentation>
    </annotation>
</element>

<complexType abstract="true" name="AbstractTunnelType">
    <complexContent>
        <extension base="con:AbstractConstructionType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the Tunnel or TunnelPart.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the Tunnel or TunnelPart.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the Tunnel or TunnelPart.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="tunnelConstructiveElement">
    <annotation>
        <documentation>Relates the constructive elements to the Tunnel or TunnelPart.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tun:TunnelConstructiveElement"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="tunnelInstallation">
    <annotation>
        <documentation>Relates the installation objects to the Tunnel or TunnelPart.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tun:TunnelInstallation"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="hollowSpace">
    <annotation>
        <documentation>Relates the hollow spaces to the Tunnel or TunnelPart.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tun:HollowSpace"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="tunnelFurniture">
<documentation>Relates the furniture objects to the Tunnel or TunnelPart.</documentation>
<complexType>
<complexContent>
<extension base="gml:AbstractFeatureMemberType">
<sequence minOccurs="0">
<element ref="tun:TunnelFurniture"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>
</complexType>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractTunnel" type="tun:ADEOfAbstractTunnelPropertyType">
<annotation>
<documentation>Augments AbstractTunnel with properties defined in an ADE.</documentation>
</annotation>
</element>

<complexType name="AbstractTunnelPropertyType">
<sequence minOccurs="0">
<element ref="tun:AbstractTunnel"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="HollowSpace" substitutionGroup="core:AbstractUnoccupiedSpace" type="tun:HollowSpaceType">
<annotation>
<documentation>A HollowSpace is a space within a Tunnel or TunnelPart intended for certain functions (e.g. transport or passage ways, service rooms, emergency shelters). A HollowSpace is bounded physically and/or virtually (e.g. by ClosureSurfaces or GenericSurfaces).</documentation>
</annotation>
</element>

<complexType name="HollowSpaceType">
<complexContent>
<extension base="core:AbstractUnoccupiedSpaceType">
<sequence>
<element minOccurs="0" name="class" type="gml:CodeType">
<annotation>
<documentation>Indicates the specific type of the HollowSpace.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the intended purposes of the HollowSpace.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
  <annotation>
    <documentation>Specifies the actual uses of the HollowSpace.</documentation>
  </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="tunnelFurniture">
  <annotation>
    <documentation>Relates the furniture objects to the HollowSpace.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="tun:TunnelFurniture"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="tunnelInstallation">
  <annotation>
    <documentation>Relates the installation objects to the HollowSpace.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
          <element ref="tun:TunnelInstallation"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfHollowSpace" type="tun:ADEOfHollowSpacePropertyType">
  <annotation>
    <documentation>Augments the HollowSpace with properties defined in an
A Tunnel represents a horizontal or sloping enclosed passage way of a certain length, mainly underground or underwater. [cf. ISO 6707-1]
<complexType name="TunnelPropertyType">
  <sequence minOccurs="0">
    <element ref="tun:Tunnel"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TunnelConstructiveElement" substitutionGroup="con:AbstractConstructiveElement" type="tun:TunnelConstructiveElementType">
  <annotation>
    <documentation>A TunnelConstructiveElement is an element of a Tunnel which is essential from a structural point of view. Examples are walls, slabs, beams.</documentation>
  </annotation>
</element>
<complexType name="TunnelConstructiveElementType">
  <complexContent>
    <extension base="con:AbstractConstructiveElementType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the TunnelConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the TunnelConstructiveElement.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the TunnelConstructiveElement.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="TunnelConstructiveElementPropertyType">
    <sequence minOccurs="0">
        <element ref="tun:TunnelConstructiveElement"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="TunnelFurniture" substitutionGroup="con:AbstractFurniture" type="tun:TunnelFurnitureType">
    <annotation>
        <documentation>A TunnelFurniture is an equipment for occupant use, usually not fixed to the tunnel. [cf. ISO 6707-1]</documentation>
    </annotation>
</element>

<complexType name="TunnelFurnitureType">
    <complexContent>
        <extension base="con:AbstractFurnitureType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the TunnelFurniture.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the TunnelFurniture.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the TunnelFurniture.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfTunnelFurniture" type="tun:ADEOfTunnelFurniturePropertyType">
                    <annotation>
                        <documentation>Augments the TunnelFurniture with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
A TunnelInstallation is a permanent part of a Tunnel (inside and/or outside) which does not have the significance of a TunnelPart. In contrast to TunnelConstructiveElement, a TunnelInstallation is not essential from a structural point of view. Examples are stairs, antennas or railings.

Indicates the specific type of the TunnelInstallation.

Specifies the intended purposes of the TunnelInstallation.

Specifies the actual uses of the TunnelInstallation.

Augments the TunnelInstallation with properties defined in an ADE.
C.15. Vegetation

The CityGML Vegetation module is defined in the XML Schema Definition file vegetation.xsd (Listing 42). The target namespace http://www.opengis.net/citygml/vegetation/3.0 is associated with this module.

Listing 42. vegetation.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:core="http://www.opengis.net/citygml/3.0"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:veg="http://www.opengis.net/citygml/vegetation/3.0"
```
The Vegetation module supports representation of vegetation objects with vegetation-specific thematic classes. CityGML's vegetation model distinguishes between solitary vegetation objects like trees, and vegetation areas which represent biotopes like forests or other plant communities.

ADEOfAbstractVegetationObject acts as a hook to define properties within an ADE that are to be added to AbstractVegetationObject.

ADEOfPlantCover acts as a hook to define properties within an ADE that are to be added to a PlantCover.

ADEOfSolitaryVegetationObject acts as a hook to define properties within an ADE that are to be added to a SolitaryVegetationObject.
<complexType name="ADEOfSolitaryVegetationObjectPropertyType">
  <sequence>
    <element ref="veg:ADEOfSolitaryVegetationObject"/>
  </sequence>
</complexType>

<element abstract="true" name="AbstractVegetationObject" substitutionGroup="core:AbstractOccupiedSpace" type="veg:AbstractVegetationObjectType">
  <annotation>
    <documentation>AbstractVegetationObject is the abstract superclass for all kinds of vegetation objects.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractVegetationObjectType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractVegetationObject" type="veg:ADEOfAbstractVegetationObjectPropertyType">
          <annotation>
            <documentation>Augments AbstractVegetationObject with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="AbstractVegetationObjectPropertyType">
  <sequence minOccurs="0">
    <element ref="veg:AbstractVegetationObject"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="PlantCover" substitutionGroup="veg:AbstractVegetationObject" type="veg:PlantCoverType">
  <annotation>
    <documentation>A PlantCover represents a space covered by vegetation.</documentation>
  </annotation>
</element>

<complexType name="PlantCoverType">
  <complexContent>
    <extension base="veg:AbstractVegetationObjectType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the
```
PlantCover.</documentation>
  </annotation>
  </element>
  <element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
    <annotation>
      <documentation>Specifies the intended purposes of the
PlantCover.</documentation>
    </annotation>
  </element>
  <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
    <annotation>
      <documentation>Specifies the actual uses of the
PlantCover.</documentation>
    </annotation>
  </element>
  <element minOccurs="0" name="averageHeight" type="gml:LengthType">
    <annotation>
      <documentation>Specifies the average height of the
PlantCover.</documentation>
    </annotation>
  </element>
  <element minOccurs="0" name="minHeight" type="gml:LengthType">
    <annotation>
      <documentation>Specifies the minimum height of the
PlantCover.</documentation>
    </annotation>
  </element>
  <element minOccurs="0" name="maxHeight" type="gml:LengthType">
    <annotation>
      <documentation>Specifies the maximum height of the
PlantCover.</documentation>
    </annotation>
  </element>
  <element maxOccurs="unbounded" minOccurs="0" name="adeOfPlantCover"
type="veg:ADEOfPlantCoverPropertyType">
    <annotation>
      <documentation>Augments the PlantCover with properties defined in an
ADE.</documentation>
    </annotation>
  </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="PlantCoverPropertyType">
  <sequence minOccurs="0">
    <element ref="veg:PlantCover"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
<complexType name="SolitaryVegetationObjectType">
<complexContent>
<extension base="veg:AbstractVegetationObjectType">
<sequence>
  <element minOccurs="0" name="class" type="gml:CodeType">
    <annotation>
      <documentation>Indicates the specific type of the SolitaryVegetationObject.</documentation>
    </annotation>
  </element>
  <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
    <annotation>
      <documentation>Specifies the intended purposes of the SolitaryVegetationObject.</documentation>
    </annotation>
  </element>
  <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
    <annotation>
      <documentation>Specifies the actual uses of the SolitaryVegetationObject.</documentation>
    </annotation>
  </element>
  <element minOccurs="0" name="species" type="gml:CodeType">
    <annotation>
      <documentation>Indicates the botanical name of the SolitaryVegetationObject.</documentation>
    </annotation>
  </element>
  <element minOccurs="0" name="height" type="gml:LengthType">
    <annotation>
      <documentation>Distance between the highest point of the vegetation object and the lowest point of the terrain at the bottom of the object.</documentation>
    </annotation>
  </element>
  <element minOccurs="0" name="trunkDiameter" type="gml:LengthType">
    <annotation>
      <documentation>Specifies the diameter of the SolitaryCityObject's trunk.</documentation>
    </annotation>
  </element>
</sequence>
</extension>
</complexContent>
</complexType>
C.16. Versioning

The CityGML Versioning module is defined in the XML Schema Definition file `versioning.xsd` (Listing 43). The target namespace `http://www.opengis.net/citygml/versioning/3.0` is associated with this module.

Listing 43. versioning.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"

```
The Versioning module supports representation of multiple versions of CityGML features within a single CityGML model. In addition, also the version transitions and transactions that lead to the different versions can be represented.

ADEOfVersion acts as a hook to define properties within an ADE that are to be added to a Version.

ADEOfVersionTransition acts as a hook to define properties within an ADE that are to be added to a VersionTransition.

Transaction represents a modification of the city model by the creation, termination, or replacement of a specific city object. While the creation of a city object also marks its first object version, the termination marks the end of
existence of a real world object and, hence, also terminates the final version of a
city object. The replacement of a city object means that a specific version of it is
replaced by a new version.</documentation>

<complexType name="TransactionType">
  <sequence>
    <element name="type" type="vers:TransactionTypeValueType">
      <annotation>
        <documentation>Indicates the specific type of the
Transaction.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="oldFeature" type="gml:ReferenceType">
      <annotation>
        <documentation>Relates to the version of the city object prior to the
Transaction.</documentation>
      </annotation>
    </element>
    <element minOccurs="0" name="newFeature" type="gml:ReferenceType">
      <annotation>
        <documentation>Relates to the version of the city object subsequent to the
Transaction.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="TransactionPropertyType">
  <sequence>
    <element ref="vers:Transaction"/>
  </sequence>
</complexType>
<simpleType name="TransactionTypeValueType">
  <annotation>
    <documentation>TransactionTypeValue enumerates the three possible types of
transactions: insert, delete, or replace.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="insert">
      <annotation>
        <documentation>Indicates that the feature referenced from the Transaction
      </annotation>
    </enumeration>
  </restriction>
</simpleType>
via the "newFeature" association has been newly created; the association "oldFeature" is empty in this case.</documentation>
</annotation>
</enumeration>
<enumeration value="delete">
  <annotation>
    <documentation>Indicates that the feature referenced from the Transaction via the "oldFeature" association ceases to exist; the association "newFeature" is empty in this case.</documentation>
  </annotation>
</enumeration>
<enumeration value="replace">
  <annotation>
    <documentation>Indicates that the feature referenced from the Transaction via the "oldFeature" association has been replaced by the feature referenced via the "newFeature" association.</documentation>
  </annotation>
</enumeration>
</restriction>
</simpleType>
<simpleType name="TransitionTypeValueType">
  <annotation>
    <documentation>TransitionTypeValue enumerates the different kinds of version transitions. "planned" and "fork" should be used in cases when from one city model version multiple successor versions are being created. "realized" and "merge" should be used when different city model versions are converging into a common successor version.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="planned">
      <annotation>
        <documentation>Indicates that the successor version of the city model represents a planning state for a possible future of the city.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="realized">
      <annotation>
        <documentation>Indicates that the predecessor version is the chosen one from a number of possible planning versions.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="historicalSuccession">
      <annotation>
        <documentation>Indicates that the successor version reflects updates on the city model over time (historical timeline). It shall only be used for at most one version transition outgoing from a city model version.</documentation>
      </annotation>
    </enumeration>
    <enumeration value="fork">
      <annotation>
        <documentation>Indicates other reasons to create alternative city model
versions, for example, when different parties are updating parts of the city model or
to reflect the results of different simulation runs.</documentation>
  </annotation>
  </enumeration>
  <enumeration value="merge">
    <annotation>
      <documentation>Indicates other reasons to converge multiple versions back
into a common city model version.</documentation>
    </annotation>
  </enumeration>
</restriction>
</simpleType>
<element name="Version" substitutionGroup="core:AbstractVersion" type="vers:VersionType">
  <annotation>
    <documentation>Version represents a defined state of a city model consisting of
the dedicated versions of all city object instances that belong to the respective city
model version. Versions can have names, a description and can be labeled with an
arbitrary number of user defined tags.</documentation>
  </annotation>
</element>
<complexType name="VersionType">
  <complexContent>
    <extension base="core:AbstractVersionType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="tag" type="string">
          <annotation>
            <documentation>Allows for adding keywords to the city model
version.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="versionMember">
          <annotation>
            <documentation>Relates to all city objects that are part of the city
model version.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfVersion"
<annotation>
  <documentation>Augments the Version with properties defined in an ADE.</documentation>
</annotation>

<complexType name="VersionPropertyType">
  <complexContent>
    <extension base="vers:Version"/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </complexContent>
</complexType>

<element name="VersionTransition" substitutionGroup="core:AbstractVersionTransition" type="vers:VersionTransitionType">
  <annotation>
    <documentation>VersionTransition describes the change of the state of a city model from one version to another. Version transitions can have names, a description and can be further qualified by a type and a reason.</documentation>
  </annotation>

  <complexType name="VersionTransitionType">
    <complexContent>
      <extension base="core:AbstractVersionTransitionType">
        <sequence>
          <element minOccurs="0" name="reason" type="string">
            <annotation>
              <documentationSpecifies why the VersionTransition has been carried out.</documentation>
            </annotation>
          </element>
          <element name="clonePredecessor" type="boolean">
            <annotation>
              <documentationIndicates whether the set of city object instances belonging to the successor version of the city model is either explicitly enumerated within the successor version object (attribute clonePredecessor=false), or has to be derived from the modifications of the city model provided as a list of transactions on the city object versions contained in the predecessor version (attribute clonePredecessor=true).</documentation>
            </annotation>
          </element>
          <element minOccurs="0" name="type" type="vers:TransitionTypeValueType">
            <annotation>
              <documentationIndicates the specific type of the VersionTransition.</documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
C.17. Water Body

The CityGML Water Body module is defined in the XML Schema Definition file waterBody.xsd.
The target namespace http://www.opengis.net/citygml/waterbody/3.0 is associated with this module.

Listing 44. waterBody.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:core="http://www.opengis.net/citygml/3.0"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:wtr="http://www.opengis.net/citygml/waterbody/3.0"
    elementFormDefault="qualified"
    targetNamespace="http://www.opengis.net/citygml/waterbody/3.0" version="3.0.0">
    <annotation>
        <documentation>The WaterBody module supports representation of the thematic aspects and 3D geometry of rivers, canals, lakes, and basins. It does, however, not inherit any hydrological or other dynamic aspects of fluid flow.</documentation>
    </annotation>
    <import namespace="http://www.opengis.net/citygml/3.0" schemaLocation="./core.xsd"/>
    <import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
    <!--XML Schema document created by ShapeChange - http://shapechange.net/-->
    <element abstract="true" name="ADEOfAbstractWaterBoundarySurface"
        substitutionGroup="gml:AbstractObject"
        type="wtr:ADEOfAbstractWaterBoundarySurfaceType">
        <annotation>
            <documentation>ADEOfAbstractWaterBoundarySurface acts as a hook to define properties within an ADE that are to be added to AbstractWaterBoundarySurface.</documentation>
        </annotation>
        <complexType abstract="true" name="ADEOfAbstractWaterBoundarySurfaceType">
            <sequence/>
        </complexType>
        <complexType name="ADEOfAbstractWaterBoundarySurfacePropertyType">
            <sequence>
                <element ref="wtr:ADEOfAbstractWaterBoundarySurface"/>
            </sequence>
        </complexType>
        <element abstract="true" name="ADEOfWaterBody"
            substitutionGroup="gml:AbstractObject" type="wtr:ADEOfWaterBodyType">
            <annotation>
                <documentation>ADEOfWaterBody acts as a hook to define properties within an ADE that are to be added to a WaterBody.</documentation>
            </annotation>
            <complexType abstract="true" name="ADEOfWaterBodyType">
                <sequence/>
            </complexType>
            <complexType name="ADEOfWaterBodyPropertyType">
                <sequence>
                    <element ref="wtr:ADEOfWaterBody"/>
                </sequence>
            </complexType>
        </element>
    </element>
</schema>
```
<element abstract="true" name="ADEOfWaterGroundSurface" substitutionGroup="gml:AbstractObject" type="wtr:ADEOfWaterGroundSurfaceType">
  <annotation>
    <documentation>ADEOfWaterGroundSurface acts as a hook to define properties within an ADE that are to be added to a WaterGroundSurface.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfWaterGroundSurfaceType">
  <sequence/>
</complexType>

<complexType name="ADEOfWaterGroundSurfacePropertyType">
  <sequence>
    <element ref="wtr:ADEOfWaterGroundSurface"/>
  </sequence>
</complexType>

<element abstract="true" name="ADEOfWaterSurface" substitutionGroup="gml:AbstractObject" type="wtr:ADEOfWaterSurfaceType">
  <annotation>
    <documentation>ADEOfWaterSurface acts as a hook to define properties within an ADE that are to be added to a WaterSurface.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="ADEOfWaterSurfaceType">
  <sequence/>
</complexType>

<complexType name="ADEOfWaterSurfacePropertyType">
  <sequence>
    <element ref="wtr:ADEOfWaterSurface"/>
  </sequence>
</complexType>

<element abstract="true" name="AbstractWaterBoundarySurface" substitutionGroup="core:AbstractThematicSurface" type="wtr:AbstractWaterBoundarySurfaceType">
  <annotation>
    <documentation>AbstractWaterBoundarySurface is the abstract superclass for all kinds of thematic surfaces bounding a water body.</documentation>
  </annotation>
</element>

<complexType abstract="true" name="AbstractWaterBoundarySurfaceType">
  <complexContent>
    <extension base="core:AbstractThematicSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractWaterBoundarySurface" type="wtr:ADEOfAbstractWaterBoundarySurfacePropertyType">
          <annotation>
            <documentation>Augments AbstractWaterBoundarySurface with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AbstractWaterBoundarySurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="wtr:AbstractWaterBoundarySurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

<element name="WaterBody" substitutionGroup="core:AbstractOccupiedSpace" type="wtr:WaterBodyType">
  <annotation>
    <documentation>A WaterBody represents significant and permanent or semi-permanent accumulations of surface water, usually covering a part of the Earth.</documentation>
  </annotation>
</element>

<complexType name="WaterBodyType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the WaterBody.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the WaterBody.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage" type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the WaterBody.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfWaterBody" type="wtr:ADEOfWaterBodyPropertyType">
          <annotation>
            <documentation>Augments the WaterBody with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element minOccurs="0" name="waterLevel" type="gml:CodeType">
    <annotation>
        <documentation>Specifies the level of the WaterSurface.</documentation>
    </annotation>
</element>

<element maxOccurs="unbounded" minOccurs="0" name="adeOfWaterSurface" type="wtr:ADEOfWaterSurfacePropertyType">
    <annotation>
        <documentation>Augments the WaterSurface with properties defined in an ADE.</documentation>
    </annotation>
</element>

<complexType name="WaterSurfacePropertyType">
    <sequence minOccurs="0">
        <element ref="wtr:WaterSurface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>
Annex D: Conceptual Model Conformance (Normative)

**NOTE**
This is where conformance with CityGML 3.0 Conceptual Model is documented.
Annex E: Codelists (Informative)

Codelists are tables of values and corresponding descriptions or definitions. Values in a specific codelist may be used for designated properties with the codelist stereotype and a gml:CodeValue datatype in the UML Conceptual Model (CM). The CityGML 3.0 CM defines 140 such values.

A gml:CodeValue datatype may have an optional codeSpace attribute. The purpose of the codeSpace attribute is to identify the codelist from which the value has been taken. Providing the codeSpace, especially with a dereferenceable identifier, allows easy access to further information detailing rules governing the use and meaning of the codelist's values and metadata. There may also be rules defining such things as the use and interpretation of values that do not appear in the codelist, the duplication of values, or any other characteristics of a codelist. All such properties and uses of a codelist or codelist value are outside the scope of this CityGML 3.0 Encoding Specification. This does not in any way restrict the ability of an authority sponsoring a codelist, as denoted by the use of a codeSpace attribute, to have application-specific codelist values, rules, interpretations, and uses in an independent specification.

The values for each of these 140 codelists are interpreted in a specific application context. In particular, they may be required, recommended, or suggested by an authority within an organization or community. As a consequence, the lists and contained values are outside the scope of standardization of the CityGML CM or any of its encodings. Since only the values themselves, not the lists or tables, appear in CityGML GML-encoded documents, the structure of codelists also lies outside the scope of CityGML CM and Encoding standards. Nevertheless, it is helpful to have some example codelists available for inspection by developers working with a specific encoding. This Annex contains example structures for GML, comma-separated-value (CSV), and JSON codelists. Supplementing these structures, OGC maintains a CityGML 3.0 codelist repository with samples of each of the 140 CityGML 3.0 codelist attributes in GML, CSV, and JSON form at https://data.ogc.org/citygml-swg/3.0.0/<Module_Name>.<Codelist_Name>.<Extension>, where <Module_Name> is the name of the Module, <Codelist_Name> is the name of the attribute in the CityGML 3.0 CM UML diagram, and Extension is “xml”, “csv”, or “json” to indicate the encoding. For example, to retrieve a sample codelist for the Vegetation Module PlantCoverClassValue, encoded as a GML 3.2 Dictionary, use the URL https://data.ogc.org/citygml-swg/CodeList_Examples_3.0.0/Vegetation.PlantCoverClassValue.xml. Note that the version number 3.0.0 is a codelist version number, not a standard version number. Once a sample codelist is published with a given version number, it will not change.

E.1. Logical Model

There is no standardized model for CityGML codelists. The following is a suggested structure based on the GML 3.2 Dictionary that meets requirements for documenting the source, currency, language, and other characteristics of the list of values. This suggested model is followed by the CityGML 3.0 codelist examples hosted by OGC. Note that many existing codelists, maintained by various organizations, are already in existence with different and permissible structures. These codelists remain valid.
E.2. Examples

E.2.1. GML

GML encoding of codelists as gml:Dictionary elements containing gml:dictionaryEntry gml:Definition elements is the expected form for defining and exchanging codelists in a GML environment. Examples of each of the 140 CityGML 3.0 codelists are provided using this encoding. Note, however, that there is no rule preventing the use of codelists encoded in some other manner. This is a common situation when fixed multipurpose codelists are recommended or required by government or other official agencies.

The following is an XML example:
E.2.2. Comma Separated Values (CSV) Structure

A suggested CSV encoding of the example logical model is illustrated with the following example with the delimiter ‘|’:

<table>
<thead>
<tr>
<th>CodeType</th>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta</td>
<td>CityGML_DataType</td>
<td>Building</td>
</tr>
<tr>
<td>Meta</td>
<td>CityGML_Namespace</td>
<td><a href="http://www.opengis.net/citygml/building/3.0">http://www.opengis.net/citygml/building/3.0</a></td>
</tr>
<tr>
<td>Meta</td>
<td>Language</td>
<td>en-UK</td>
</tr>
<tr>
<td>Meta</td>
<td>CityGML_Version</td>
<td>3.0</td>
</tr>
<tr>
<td>Code</td>
<td>1000</td>
<td>flat roof</td>
</tr>
<tr>
<td>Code</td>
<td>3100</td>
<td>saddle roof</td>
</tr>
</tbody>
</table>
A suggested JSON encoding of the example logical model for codelists has a JSON-Schema 2019-09 definition as follows:

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "Dictionary": {
            "type": "object",
            "properties": {
                "@id": {
                    "type": "string"
                },
                "metaDataProperty": {
                    "type": "object",
                    "properties": {
                        "CodeListMetaData": {
                            "type": "object",
                            "properties": {
                                "dataType": {
                                    "type": "string"
                                },
                                "namespace": {
                                    "type": "string"
                                },
                                "language": {
                                    "type": "string"
                                },
                                "authority": {
                                    "type": "string"
                                },
                                "version": {
                                    "type": "string"
                                }
                            },
                            "required": ["dataType", "namespace", "language", "authority", "version"]
                        }
                    },
                    "required": ["CodeListMetaData"]
                }
            },
            "required": ["CodeListMetaData"]
        }
    },
    "required": ["Dictionary"]
}
```
"description": {
    "type": "string"
},
"identifier": {
    "type": "object",
    "properties": {
        "@codeSpace": {
            "type": "string"
        },
        "#text": {
            "type": "string"
        }
    },
    "required": [
        "@codeSpace",
        "#text"
    ]
},
"dictionaryEntry": {
    "type": "array",
    "items": [
        {
            "type": "object",
            "properties": {
                "Definition": {
                    "type": "object",
                    "properties": {
                        "@id": {
                            "type": "string"
                        },
                        "description": {
                            "type": "string"
                        }
                    },
                    "identifier": {
                        "type": "object",
                        "properties": {
                            "@codeSpace": {
                                "type": "string"
                            },
                            "#text": {
                                "type": "string"
                            }
                        },
                        "required": [
                            "@codeSpace",
                            "#text"
                        ]
                    }
                },
                "name": {
                    "type": "string"
                }
            }
        }
    ]
}


```
},
"required": [  
  
  ],
}
"required": [  
  
  ],
}
"type": "object",
"properties": {
  "Definition": {
    "@id": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "identifier": {
      "type": "object",
      "properties": {
        "@codeSpace": {
          "type": "string"
        },
       ="#text": {
          "type": "string"
        }
      },
      "required": [  
        "@codeSpace",
       ="#text"
      ]
    },
    "name": {
      "type": "string"
    }
  },
  "required": [  
    "@id",
    "description",
    "identifier",
    "name"
  ]
}
```
The following is the previous example in the suggested JSON encoding:
"Dictionary": {
  "@id": "roofTypes",
  "metaDataProperty": {
    "CodeListMetaData": {
      "dataType": "RoofTypeValue",
      "namespace": "http://www.opengis.net/citygml/building/3.0",
      "language": "en",
      "authority": "Some Organization",
      "version": "3.0"
    }
  },
  "description": "Roof type values",
  "identifier": {
    "@codeSpace": "https://ogc.org/citygml/3.0/codelists/gml/rooftypes",
    "#text": "RoofTypeValue"
  },
  "dictionaryEntry": [
    {
      "Definition": {
        "@id": "id1",
        "description": "roof primarily a single plane, not necessarily level",
        "identifier": {
          "@codeSpace": "https://ogc.org/citygml/3.0/codelists/gml/rooftypes",
          "#text": "1000"
        },
        "name": "flat roof"
      }
    },
    {
      "Definition": {
        "@id": "id2",
        "description": "a roof that has a ridge and two gables",
        "identifier": {
          "@codeSpace": "https://ogc.org/citygml/3.0/codelists/gml/rooftypes",
          "#text": "3100"
        },
        "name": "saddle roof"
      }
    }
  ]
}
Annex F: Use of GML 3.3 (Informative)

The CityGML 3.0 GML schemas have been derived based on the OGC GML 3.2 standard. This means that CityGML GML instance documents must be created and exchanged in the GML version 3.2.

For certain applications, however, it might be useful to create CityGML GML instance documents in the GML version 3.3. GML 3.3 complements GML 3.2 with concepts such as compact encodings of geometry types, linear referencing concepts (e.g., linear element, distance along, linear referencing methods), TINs, and encoding rules for association classes.

Please note that software compliant to this standard might not be able to read the additional GML 3.3 concepts. This depends on whether the software only supports GML 3.2 or also GML 3.3.

F.1. Using GML 3.3 concepts in a CityGML instance document

To be able to create CityGML instance documents based on GML 3.3, those GML 3.3 XML schemas that define the required concepts need to be referenced in the XML root element <CityModel> of CityGML GML instance documents. This is illustrated in Listing 45. The RoofSurface of the building is to be represented by a SimplePolygon which is a compact encoding of a Polygon defined in the GML 3.3 XML schema http://www.opengis.net/gml/3.3/ce. Thus, this XML schema is referenced in the root element by indicating its namespace and schema location.
Listing 45. Use of GML 3.3 concepts in a CityGML GML instance document.

```xml
<CityModel xmlns:gmlce="http://www.opengis.net/gml/3.3/ce"
            xsi:schemaLocation="http://www.opengis.net/gml/3.3/ce
http://schemas.opengis.net/gml/3.3/geometryCompact.xsd">
  <cityObjectMember>
    <bldg:Building gml:id="DEBY_LOD2_5744682">
      ....
      <boundary>
        <con:RoofSurface>
          <lod2MultiSurface>
            <gml:MultiSurface gml:id="ms1">
              <gml:surfaceMember>
                <gmlce:SimplePolygon gml:id="sp1">
                  <gml:posList>
                    ... coordinates ...
                  </gml:posList>
                </gmlce:SimplePolygon>
              </gml:surfaceMember>
            </gml:MultiSurface>
          </lod2MultiSurface>
        </con:RoofSurface>
      </boundary>
    </bldg:Building>
  </cityObjectMember>
</CityModel>
```

F.2. Assigning textures to compact surface geometries

GML 3.3 introduces compact encodings for commonly used surface geometries such as `SimplePolygon`, `SimpleRectangle`, and `SimpleTriangle`. These simplified encodings are logically equivalent to a surface with a single `gml:LinearRing` as exterior boundary and without interior boundaries. Due to the simplification, the encodings do not use a `gml:LinearRing` element anymore that could be referenced from the ring property of a `TexCoordList` element.

To still be able to assign texture coordinates to simplified surface geometries of GML 3.3, the following recommendation should be followed.

<table>
<thead>
<tr>
<th>Recommendation 3</th>
<th>/rec/appearance/parameterizedtexture-gml-3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigning texture coordinates to a compact surface geometry of GML 3.3 using ParameterizedTexture elements should adhere to the following recommendations:</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>The <code>textureCoordinates</code> property SHOULD be used in the same way as for a normal surface geometry and as defined by /req/appearance/parameterizedtexture. The number of 2D points in the <code>textureCoordinates</code> element therefore must be identical with the number of 3D points of the simplified geometry.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>B</td>
<td>In contrast to a <code>gml:LinearRing</code>, the last point of a compact geometry does not have to repeat the first point. If it is repeated though, it SHOULD also be repeated for the texture coordinates.</td>
</tr>
<tr>
<td>C</td>
<td>The <em>ring</em> property SHOULD be an empty element.</td>
</tr>
</tbody>
</table>
Annex G: Implementation Decisions (Informative)

G.1. Encoding of association classes

G.1.1. GML encoding of association classes according to ISO 19136-2

ISO 19136-2 defines an encoding rule for association classes. The encoding rule comprises several steps for deriving a GML encoding from association classes which will be explained in the following based on the UML model provided in Figure 29.

Figure 29. UML object diagram for the building in Figure 12.

The UML model in Figure 29 defines association class X with an association between source class A and target class B. The following steps are defined by the encoding rule, resulting in the UML model shown in Figure 30:

- Association class X is converted into intermediate class X. The new intermediate class has the same name, stereotype, tagged values, constraints, attributes, and relationships as the original association class.

- The association between source class A and target class B is replaced by two associations, association 1 between the classes A and X, and association 2 between the classes X and B.

- The association ends at class X of association 1 and at class B of association 2 receive the role name, navigability, stereotype, and tagged values of the association end at the original target class B. In addition, the association end at class X receives the multiplicity of the association end at the original target class B. The association end at class B receives multiplicity 1.

- The association ends at class A of association 1 and at class X of association 2 receive the role name, navigability, stereotype, and tagged values of the association end at the original source class A. In addition, the association end at class X receives the multiplicity of the association end at the original source class A. The association end at class A receives multiplicity 1.
The encoding rule can be applied to association classes that are defined with unidirectional, bidirectional or unspecified navigability. Correspondingly, the UML model that results from the encoding rule will also have the associations defined unidirectional, bidirectional or unspecified. The examples here focus on unidirectional associations, since all associations in the CityGML 3.0 Conceptual Model are defined with unidirectional navigability.

### G.1.2. Encoding of the tagged value “inlineOrByReference” within association classes

The tagged value “inlineOrByReference” from ISO 19136-1 is commonly used for associations to define how a feature (the so-called referenced feature) that is referenced by another feature (the so-called referencing feature) is to be represented in GML instance documents. Three different values are defined for this tagged value:

- **inline**: the referenced feature is embedded inside the referencing feature
- **byReference**: the referenced feature is provided elsewhere in the same or an external GML instance document and is referenced from the referencing feature using XLink
- **inlineOrByReference**: both representations, i.e. inline and byReference, are possible and, in addition, a mixture of both representations.

When making use of this tagged value in association classes, the encoding rule described above will add this tagged value to the corresponding association ends of association 1 and 2 after having created the intermediate class. Figure 31 and Figure 32 illustrate this. Figure 31 assumes that for the association end at class B the tagged value “inlineOrByReference” is set to the value “inline”. After applying the encoding rule, both the association ends at class X and at class B, will exhibit the value “inline” as is shown in Figure 32. Similarly, when the association in Figure 31 will have the value “byReference” or “inlineOrByReference”, both associations in Figure 4 will exhibit the value “byReference” or “inlineOrByReference”, respectively.

Figure 30. UML model resulting from applying the encoding rule.

Figure 31. Association class with the tagged value “inlineOrByReference” set to “inline”.

Figure 32. Association class with the tagged value “inlineOrByReference” set to “byReference”.

Figure 33. Association class with the tagged value “inlineOrByReference” set to “inlineOrByReference”.
Using this tagged value means that three different encodings can be obtained depending on which of the three values is set. These different encodings are illustrated in the following by three different GML instance documents. The source class A, the target class B, and the intermediate class X are represented by corresponding XML elements <A>, <B>, and <X>. The associations between the classes A and X and between X and B are both represented by the property element <role>. For illustration purposes, the GML instance documents do not contain root elements and namespaces.

The first GML instance document (Listing 46) is obtained when setting the tagged value to “inline”. Here, element X needs to be provided inline element A and element B inline element X.

**Listing 46. GML instance document for the value “inline”.

```xml
<A gml:id="f1">
  <role>
    <X gml:id="f3">
      <role>
        <B gml:id="f2">
          </B>
        </role>
    </X>
  </role>
</A>
```

The second GML instance document (Listing 47) results from setting the tagged value to “byReference”. Here, element A references element X and element X references element B using XLink.

**Listing 47. GML instance document for the value “byReference”.

```xml
<A gml:id="f1">
  <role xlink:href="#f3"/>
</A>
<X gml:id="f3">
  <role xlink:href="#f2"/>
</X>
<B gml:id="f2">
</B>
```

The third GML instance document (Listing 48) is obtained when the tagged value is set to “inlineOrByReference”. Here, the inline and byReference representations are combined, i.e., element X is provided inline element A and element B is referenced by element X using XLink. Alternatively, it is also possible that element X is referenced by element A and element B is

---

Figure 32. “inlineOrByReference” settings after applying the encoding rule.
provided inline element X. In addition, also the GML instances as shown above for “inline” and “byReference” can be represented with the value “inlineOrByReference”.

Listing 48. GML instance document for the value “inlineOrByReference”.

```xml
<A gml:id="f1">
  <role>
    <X gml:id="f3">
      <role xlink:href="f2"/>
    </X>
  </role>
</A>

<B gml:id="f2">
</B>
```

G.1.3. Restricting the combination of inline and byReference representations in the GML encoding using an Implementation Model

As described above, four different instance representations are possible in the case of the “inlineOrByReference” value. This behaviour is not desired in the GML encoding of the CityGML 3.0 Conceptual Model, as it allows for too many possibilities of how to reference features and, thus, needs to be restricted.

In the CityGML 3.0 Conceptual Model, this setting affects two association classes, `CityObjectRelation` in the Core module and `Role` in the CityObjectGroup module. For both, the only desired way of representing them in GML instance documents is the structure shown in Listing 48. This structure can be specified in a UML model as shown in Figure 33. After converting the association class into an intermediate class, the tagged value of association 1 needs to receive the value “inline” and the tagged value of association 2 the value “byReference”. (Please note: The CityGML 3.0 Conceptual Model also defines the association class `TextureAssociation` in the Appearance module. This association class, however, is not affected here, because it makes use of the value “inline” for which the encoding is correct.)

![Figure 33. Desired settings for the value “inlineOrByReference” after applying the encoding rule.](image)

In order to obtain the restricted structure in GML instance documents, an Implementation Model is created prior to the GML encoding of the CityGML 3.0 Conceptual Model. Within the Implementation Model, the association classes are manually converted into intermediate classes and the tagged values of the associations are set as shown in Figure 33. This means that the encoding will directly be performed on the Implementation Model. This solution guarantees for representing references between features according to Listing 48 in GML instance documents and it can be applied directly without any changes to the conversion tools.

In the following, the solution will be exemplified based on the association class CityObjectRelation.
This association class can be used to specify relationships between different features, e.g. it can be expressed that the WallSurface of one building shares the Polygon geometry with the WallSurface of a second Building. The association class is illustrated in Figure 34, the tagged value is set to “inlineOrByReference”.

Figure 34. Association class CityObjectRelation.

Within the Implementation Model, this association class is manually converted into an intermediate class and the tagged values of the associations are set as illustrated in Figure 35. Afterwards, the XML schemas are derived from the Implementation Model.

Figure 35. Association class CityObjectRelation represented as intermediate class.

Within a GML instance document, specifying that the WallSurfaces of two buildings share the same geometry is then restricted to the structure as shown in Listing 49. The source and target classes are represented by two XML elements <WallSurface> and the intermediate class by the element <CityObjectRelation>. The element WallSurface of building 1 provides the element CityObjectRelation inline, whereas the element CityObjectRelation references the WallSurface element of building 2 using XLink. In the same way, building 2 provides the element CityObjectRelation inline its WallSurface, and the CityObjectRelation references the WallSurface of building 1 using Xlink.
Listing 49. GML instance document for the association class CityObjectRelation.

```xml
<bldg:Building gml:id="bldg_1">
  <boundary>
    <con:WallSurface gml:id="bldg_1_ws_2">
      <relatedTo>
        <CityObjectRelation>
          <relationType>shared</relationType>
          <relatedTo xlink:href="#bldg_2_ws_4"/>
        </CityObjectRelation>
        <relatedTo>
          <CityObjectRelation>
            <relationType>equal</relationType>
            <relatedTo xlink:href="#bldg_1_ws_2"/>
          </CityObjectRelation>
          <lod2MultiSurface> ... </lod2MultiSurface>
        </relatedTo>
      </relatedTo>
    </con:WallSurface>
  </boundary>
</bldg:Building>
<bldg:Building gml:id="bldg_2">
  <boundary>
    <con:WallSurface gml:id="bldg_2_ws_4">
      <relatedTo>
        <CityObjectRelation>
          <relationType>shared</relationType>
          <relatedTo xlink:href="#bldg_2_ws_4"/>
        </CityObjectRelation>
        <relatedTo>
          <CityObjectRelation>
            <relationType>equal</relationType>
            <relatedTo xlink:href="#bldg_1_ws_2"/>
          </CityObjectRelation>
          <lod2MultiSurface> ... </lod2MultiSurface>
        </relatedTo>
      </relatedTo>
    </con:WallSurface>
  </boundary>
</bldg:Building>
```
## Annex H: Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Editor</th>
<th>Primary clauses modified</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-12-15</td>
<td>0.1</td>
<td>T. Kutzner</td>
<td>all</td>
<td>initial version</td>
</tr>
<tr>
<td>2022-02-01</td>
<td>0.2</td>
<td>T. Kutzner, S. Smyth</td>
<td>Conformance, Codelist Annex</td>
<td>Most sections are complete, at least in draft form.</td>
</tr>
</tbody>
</table>
Annex I: Glossary

conformance test class
The set of conformance test modules that must be applied to receive a single certificate of conformance
[OGC 08-131r3, definition 4.4]

feature
An abstraction of real world phenomena
[ISO 19101-1:2014, definition 4.1.11]

feature attribute
A characteristic of a feature
[ISO 19101-1:2014, definition 4.1.12]

feature type
A class of features having common characteristics
[ISO 19156:2011, definition 4.7]

measurement
A set of operations having the object of determining the value of a quantity

model
An abstraction of some aspects of reality
[ISO 19109:2015, definition 4.15]

observation
The act of measuring or otherwise determining the value of a property
[ISO 19156:2011, definition 4.11]

observation procedure
A method, algorithm or instrument, or system of these, which may be used in making an observation
[ISO 19156:2011, 4.12]

observation result
An estimate of the value of a property determined through a known observation procedure
[ISO 19156:2011, 4.14]

property
A facet or attribute of an object referenced by a name.
[ISO 19143:2010, definition 4.21]

requirements class
The aggregate of all requirement modules that must all be satisfied to satisfy a conformance test class
[OGC 08-131r3, definition 4.19]

schema
The formal description of a model
[ISO 19101-1:2014, definition 4.1.34]

Sensor
A type of observation procedure that provides the estimated value of an observed property at its output
[OGC 08-094r1, definition 4.5]

Standardization Target
An entity to which some requirements of a standard apply
[OGC 08-131r3, definition 4.23]

Timeseries
A sequence of data values which are ordered in time
[OGC 15-043r3]

Universe of discourse
View of the real or hypothetical world that includes everything of interest
[ISO 19101-1:2014, definition 4.1.38]

Version
Particular variation of a spatial object
[INSPIRE Glossary]

I.1. Abbreviated Terms

- 2D Two Dimensional
- 3D Three Dimensional
- ALKIS German National Standard for Cadastral Information
- ATKIS German National Standard for Topographic and Cartographic Information
- BIM Building Information Modeling
- COLLADA Collaborative Design Activity
- 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
- ISO/TC 211 ISO Technical Committee 211
- LOD Level of Detail
- MQTT
• OASIS  Organisation for the Advancement of Structured Information Standards
• OGC  Open Geospatial Consortium
• SIG 3D  Special Interest Group 3D of the GDI-DE
• TIC  Terrain Intersection Curve
• TIN  Triangulated Irregular Network
• UML  Unified Modeling Language
• URI  Uniform Resource Identifier
• W3C  World Wide Web Consortium
• X3D  Open Standards XML-enabled 3D file format of the Web 3D Consortium
• XML  Extensible Markup Language
• xAL  OASIS extensible Address Language
Annex J: Bibliography

- Open Geospatial Consortium: The Specification Model—A Standard for Modular specifications, OGC 08-131


• Vretanos, P. A. 2010: OpenGIS Web Feature Service 2.0 Interface Standard, Open Geospatial Consortium. OGC Doc. 09-025r1

• OASIS MQTT Technical Committee: MQTT Version 5.0 Standard, OASIS, March 7, 2019, Available from OASIS.

• Reed, C., Belayneh T.: OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package Format Specification, Open Geospatial Consortium, Available from OGC Doc. 17-014r7

• [(3dtiles_citation, OGC 3D Tiles)]Cozzi, P., Lilley, S., Getz, G. OGC 3D Tiles Specification 1.0 Open Geospatial Consortium, Available from OGC Doc. 18-053r2

• Burggraf, D.: OGC KML 2.3, Open Geospatial Consortium, Available from OGC Doc. 12-007r2


• Bhatia, S.,Cozzi, P., Knyazev, A., Parisi, T.: The GL Transmission Format (gITF), The Khronos
Group, Available from https://www.khronos.org/gltf.