

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2023-01-18>

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

Internal reference number of this OGC® document: 21-006

Version: 1.0.2

Category: OGC® Implementation Specification

Editors: Tatjana Kutzner, Carl Stephen Smyth, Claus Nagel, Volker Coors, Diego Vinasco-Alvarez, Nobuhiro Ishimaru, Zhihang Yao, Charles Heazel, Thomas H. Kolbe

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

Copyright notice

Copyright © 2023 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

License Agreement

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

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

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

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

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

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

Table of Contents

1. Scope	9
2. Conformance	10
2.1. Implementation Specifications	10
2.2. Conformance Classes	10
3. References	11
4. Terms and Definitions	12
5. Conventions	15
5.1. Identifiers	15
5.2. UML Notation	15
5.3. XML Notation	15
6. Requirements	16
6.1. Global Requirements	16
6.2. Core	31
6.3. Appearance	36
6.4. Bridge	40
6.5. Building	44
6.6. CityFurniture	48
6.7. CityObjectGroup	50
6.8. Construction	52
6.9. Dynamizer	57
6.10. Generics	59
6.11. LandUse	62
6.12. PointCloud	63
6.13. Relief	65
6.14. Transportation	66
6.15. Tunnel	71
6.16. Vegetation	75
6.17. Versioning	77
6.18. WaterBody	79
7. Media Types	82
Annex A: Conformance Class Abstract Test Suite (Normative)	83
A.1. Conformance Class Global	83
A.2. Conformance Class Core	84
A.3. Conformance Class Appearance	85
A.4. Conformance Class Bridge	87
A.5. Conformance Class Building	87
A.6. Conformance Class CityFurniture	88
A.7. Conformance Class CityObjectGroup	89

A.8. Conformance Class Construction	90
A.9. Conformance Class Dynamizer	90
A.10. Conformance Class Generics	91
A.11. Conformance Class LandUse	92
A.12. Conformance Class PointCloud	92
A.13. Conformance Class Relief	92
A.14. Conformance Class Transportation	93
A.15. Conformance Class Tunnel	94
A.16. Conformance Class Vegetation	94
A.17. Conformance Class Versioning	95
A.18. Conformance Class WaterBody	96
Annex B: Examples (Informative)	98
B.1. Dynamizer Examples	98
B.2. Point Cloud Examples	118
B.3. Versioning Examples	122
B.4. Address Examples	130
B.5. Application Domain Extension (ADE) Examples	132
Annex C: XML Schema (Normative)	152
C.1. Core	152
C.2. Appearance	184
C.3. Bridge	197
C.4. Building	208
C.5. City Furniture	226
C.6. City Object Group	228
C.7. Construction	231
C.8. Dynamizer	255
C.9. Generics	272
C.10. Land Use	282
C.11. Point Cloud	284
C.12. Relief	286
C.13. Transportation	294
C.14. Tunnel	319
C.15. Vegetation	330
C.16. Versioning	335
C.17. Water Body	341
Annex D: CityGML 3.0 Conceptual Model Conformance (Normative)	346
D.1. Conformance of this Implementation Specification to the CityGML-3.0 CM	346
Annex E: Codelists (Informative)	347
E.1. Logical Model	347
E.2. Examples	348
Annex F: Use of GML 3.3 (Informative)	355

F.1. Using GML 3.3 concepts in a CityGML instance document	355
F.2. Assigning textures to compact surface geometries	356
Annex G: Implementation Decisions (Informative).....	358
G.1. Encoding of association classes	358
G.2. Encoding of list types	363
Annex H: Revision History.....	364
Annex I: Glossary	365
I.1. Abbreviated Terms	366
Annex J: Bibliography	368

i. Abstract

The CityGML 3.0 Part 2: GML Encoding Standard presents the implementation-dependent GML encoding of the concepts defined by the CityGML 3.0 Part 1: Conceptual Model (CM) Standard ([OGC 20-010](#)). The GML Encoding is compliant to GML versions 3.2 and 3.3, which is specified by ISO 19136. This encoding can be used to store and exchange 3D city models in the GML format as data sets or via web services. 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.

This Part 2 of the Standard defines how the concepts developed in Part 1 are realized using XML and GML technologies. The GML Encoding represents a full mapping of the Conceptual Model and is derived fully automatically from the UML model following the UML-to-GML encoding rules as defined by ISO 19136. [Table 1](#) maps requirements classes from the CityGML 3.0 Conceptual Model into the implementation details documented in this standard.

Table 1. Conceptual Model Mapping

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

ii. Keywords

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

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

iii. Preface

In order to achieve consensus on the basic entities, attributes, and relations of a 3D city model, a UML Conceptual Model, CityGML 3.0 Part 1, was approved as an OGC standard ([OGC 20-010](#)) in March, 2021. This model provides a unifying conceptual basis for city model encoding standards. This CityGML 3.0 Part 2: 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 GML standards version 3.2 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 Part 1: 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 Part 1: 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. It includes requirements that can be checked in a formalized way to verify the correctness of a CityGML implementation. 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 GML Encoding Standard is a standardization target of the CityGML 3.0 Part 1: 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
- HFT Stuttgart
- OpenSitePlan
- Ordnance Survey
- Technical University of Munich
- Virtual City Systems

v. Submitters

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

Name	Affiliation
Volker Coors	HFT Stuttgart, Germany
Charles Heazel	Heazeltech LLC, USA
Thomas H. Kolbe	Chair of Geoinformatics, Technical University of Munich, Germany
Tatjana Kutzner	Chair of Geoinformatics, Technical University of Munich, Germany
Claus Nagel	Virtual City Systems, Germany
Carl Stephen Smyth	OpenSitePlan, USA

vi. Participants in development

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

Table 2. Participants in Development

Name	Institution
Christof Beil	Chair of Geoinformatics, Technical University of Munich, Germany
Kanishk Chaturvedi	Chair of Geoinformatics, Technical University of Munich, Germany
Dean Hintz	Safe Software, Canada
Peter Parslow	Ordnance Survey, Great Britain

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. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, users are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

- IETF: RFC 2045 & 2046, Multipurpose Internet Mail Extensions (MIME). (November 1996),
- IETF: RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. (January 2005)
- ISO: ISO 19109:2015, Geographic Information – Rules for Application Schemas
- 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
- OGC: The OpenGIS® Abstract Specification Topic 5: Features, OGC document 08-126
- OGC: The OpenGIS™ Abstract Specification Topic 8: Relationships Between Features, OGC document 99-108r2
- OGC: The OpenGIS™ Abstract Specification Topic 10: Feature Collections, OGC document 99-110
- OGC: OGC 20-010, OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard, v3.0, 2021.
- OGC: OGC 07-036r1, OpenGIS® Geography Markup Language (GML) Encoding Standard, v3.2.2, 2007.

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 07-036r1](#), 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.

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-global	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	-

6.1.1. Use of GML 3.2

Requirement 1	/req/global/gml
XML instance documents claiming conformance to this specification shall validate against the XML schema files provided in Appendix C .	
Note: This means that they have to be valid GML 3.2 documents. They may use certain aspects of GML 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 the values of the *gml:id* and the *gml:identifier* might also be identical, for example, when only a first version of the object exists. Please also 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.

UML attributes to provide (top-level) features with identity	Mapping to corresponding GML concepts	Definition
featureID	XML attribute <i>gml:id</i>	Specifies the unique identifier of the feature that is valid in the instance document within which it occurs.
identifier	XML element <i>gml:identifier</i>	Specifies the global identifier of the feature. The identifier must be globally valid, unique and identical over different feature versions.

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

```
<bldg:Building gml:id="B1020_t1">
  <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`.

```

<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](#).

Listing 3. Complex structure of CityGML features.

```
<bldg:Building gml:id="BU_234">
  <core:creationDate>2019-09-24T00:00:00</core:creationDate>
  <core:boundary>
    <con:WallSurface gml:id="WS_21">
      <core:lod2MultiSurface>
        <gml:MultiSurface gml:id="MS_21_1">
          ...
          </gml:MultiSurface>
        </core:lod2MultiSurface>
      </con:WallSurface>
    </core:boundary>
    <core:lod2Solid>
      <gml:Solid gml:id="S_1"> ... </gml:Solid>
    </core:lod2Solid>
    <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>1000</bldg:function>
    <bldg:buildingRoom>
      <bldg:BuildingRoom gml:id="BR_3">
        ...
      </bldg:BuildingRoom>
    </bldg:buildingRoom>
  </bldg:Building>
```

6.1.4. Rules for linking features and geometries that are shared by multiple 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.	
<p>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.</p> <p>Note 2: ImplicitGeometry elements of different top-level features MAY reference the same geometry using XLinks. This is an exception to this requirement.</p>	

Requirement 3	/req/global/referencinggeometries2
Referencing geometries of spaces and space boundaries is subject to the following restrictions:	
A	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.
B	Space boundaries SHALL NOT reference geometries of a space using XLinks.
C	Rules A and B apply for all spaces and space boundaries that are children of the same top-level feature.
<p>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.</p> <p>Note 2: If the space does not have space boundaries, then the geometry is stored with the space and no XLinks are required.</p>	

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).

Requirement 5 /req/global/referencinggeometries4	
Referencing shared geometries between top-level features using CityObjectRelations is subject to the following restrictions:	
A	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).
B	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.
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.

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

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

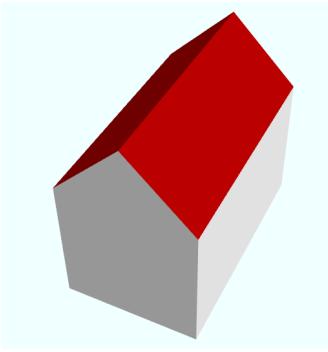


Figure 1. Simple building with solid geometry and boundary surfaces.

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 on the [CityGML 3.0 GML Encoding GitHub](#).

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.

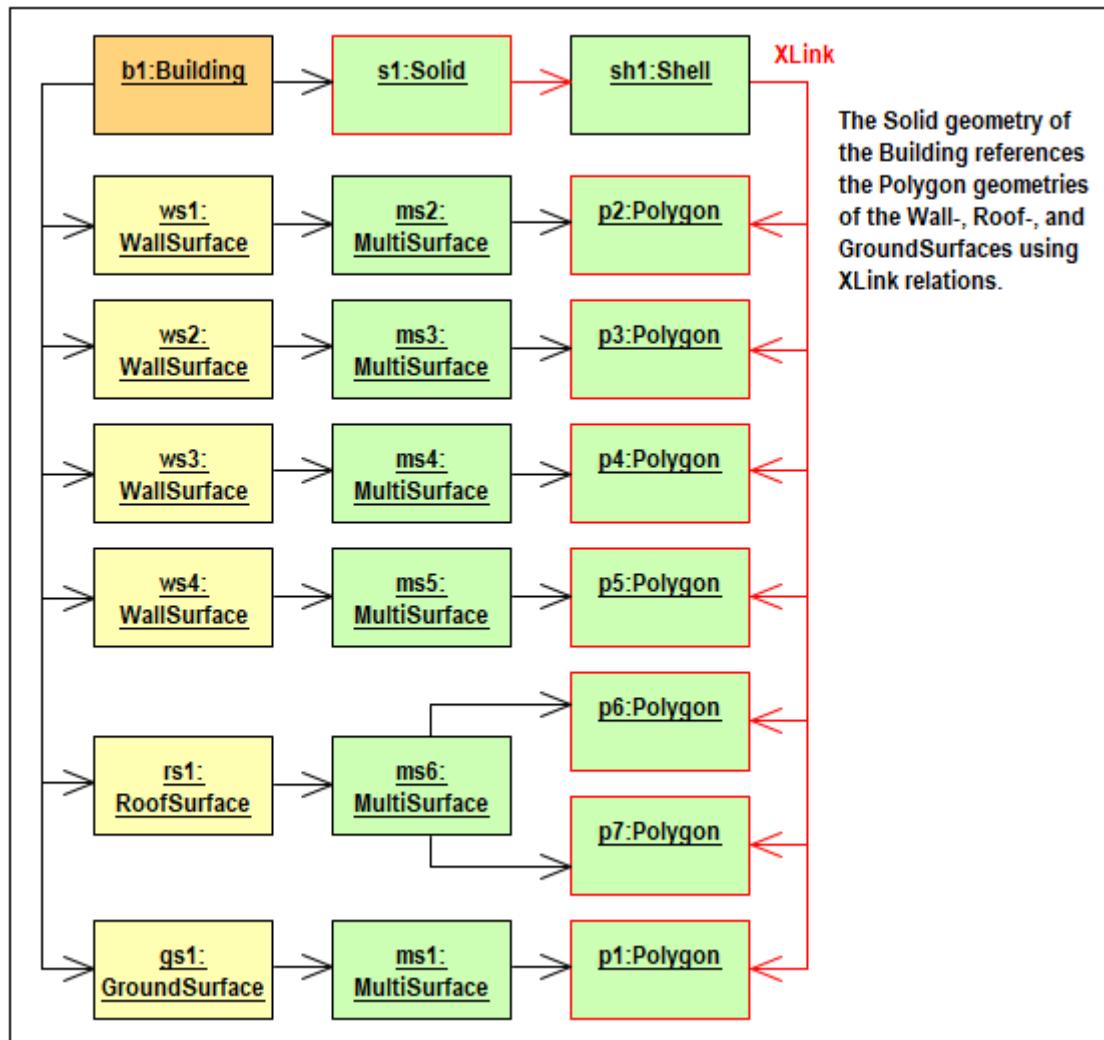


Figure 2. UML object diagram for the building in Figure 1.

Example 2: Building with roof overhangs

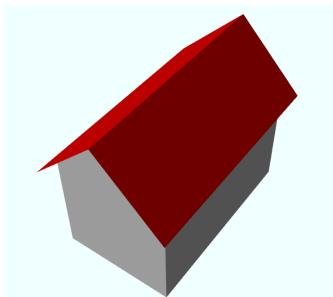


Figure 3. Building with roof overhangs.

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 on the [CityGML 3.0 GML Encoding GitHub](#).

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.

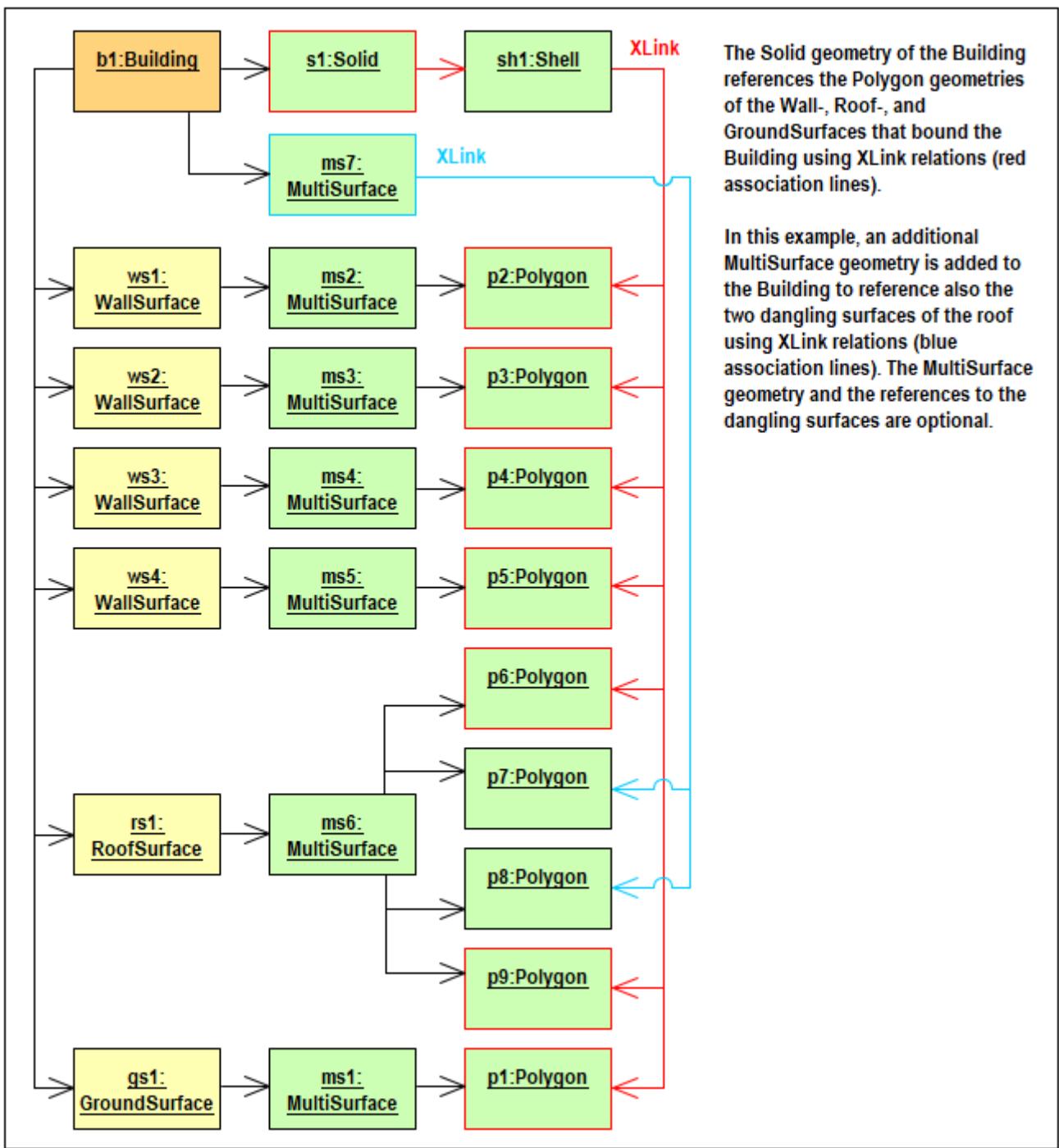


Figure 4. UML object diagram for the building in Figure 3.

Example 3: Building with BuildingInstallation

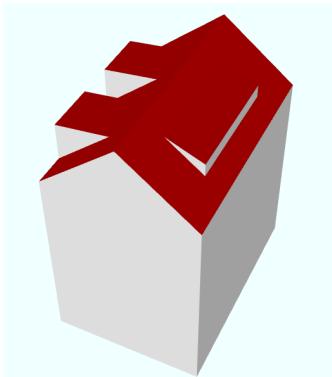


Figure 5. Building with building installation.

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. Moreover, in this example, the Solid geometry of the building has to additionally reference the Polygon geometries that represent the space boundaries of the building installation using XLink to ensure a closed and watertight volume geometry. In general, however, references to the geometries of a nested space are optional, and it is allowed to not reference them in accordance with Requirement [/req/global/referencinggeometries2](#).

The GML file is available on the [CityGML 3.0 GML Encoding GitHub](#).

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.

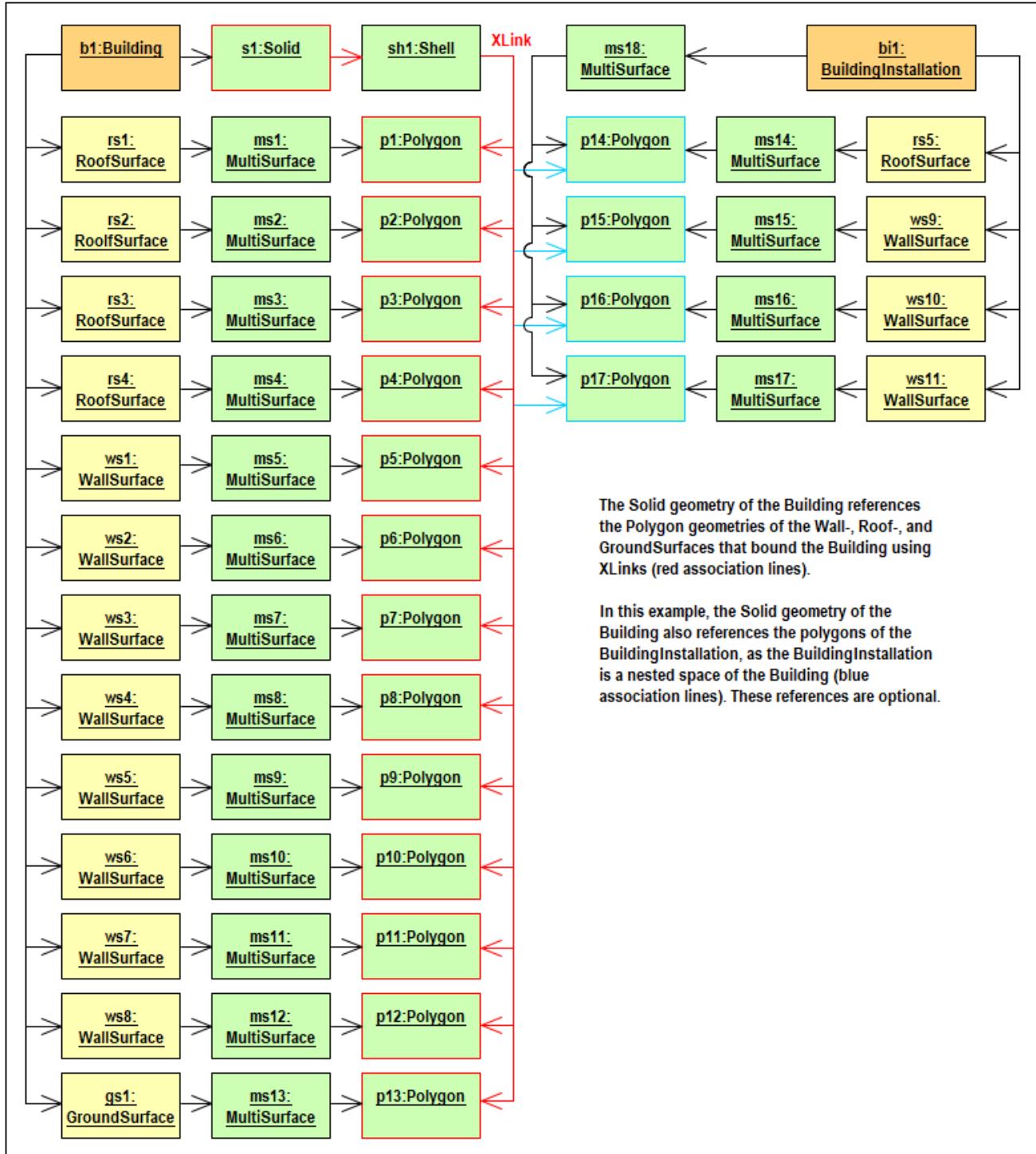


Figure 6. UML object diagram for the building in Figure 5.

Example 4: Two buildings with shared boundary surface

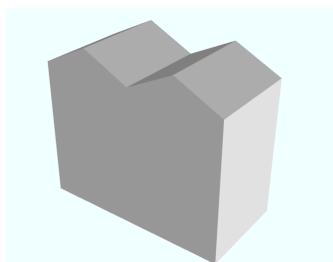


Figure 7. Two buildings with shared boundary surface.

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 on the [CityGML 3.0 GML Encoding GitHub](#).

The Buildings from the GML file are illustrated in the object diagram in [Figure 8](#). The CityObjectRelation is highlighted in red.

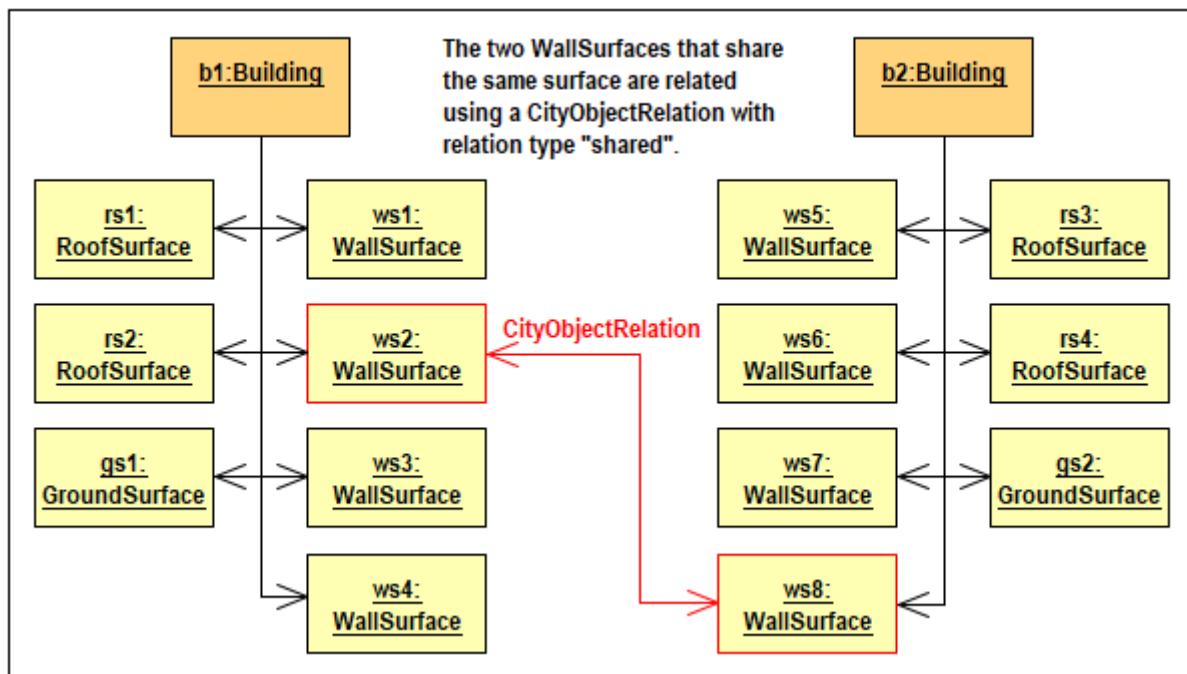


Figure 8. UML object diagram for the building in [Figure 7](#).

Example 5: Road crossing a Bridge

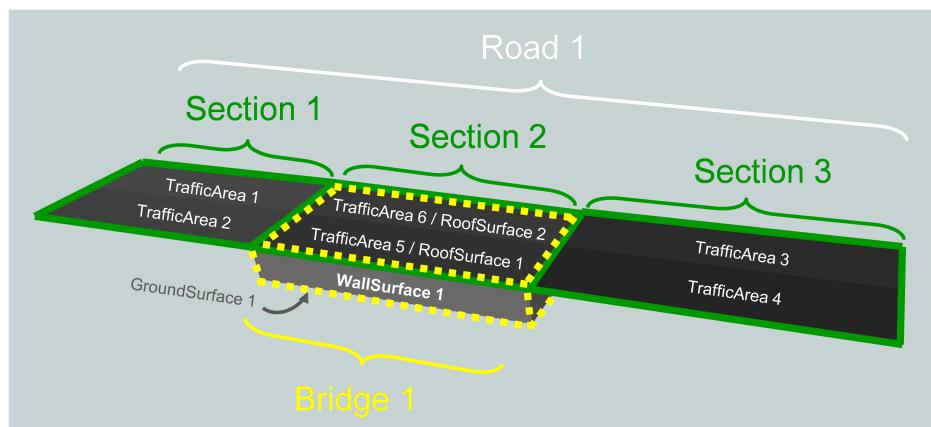


Figure 9. Shared surfaces between a road and a bridge.

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 on the [CityGML 3.0 GML Encoding GitHub](#).

The Road and Bridge from the GML file are illustrated in the object diagram in [Figure 10](#). The CityObjectRelations are highlighted in red.

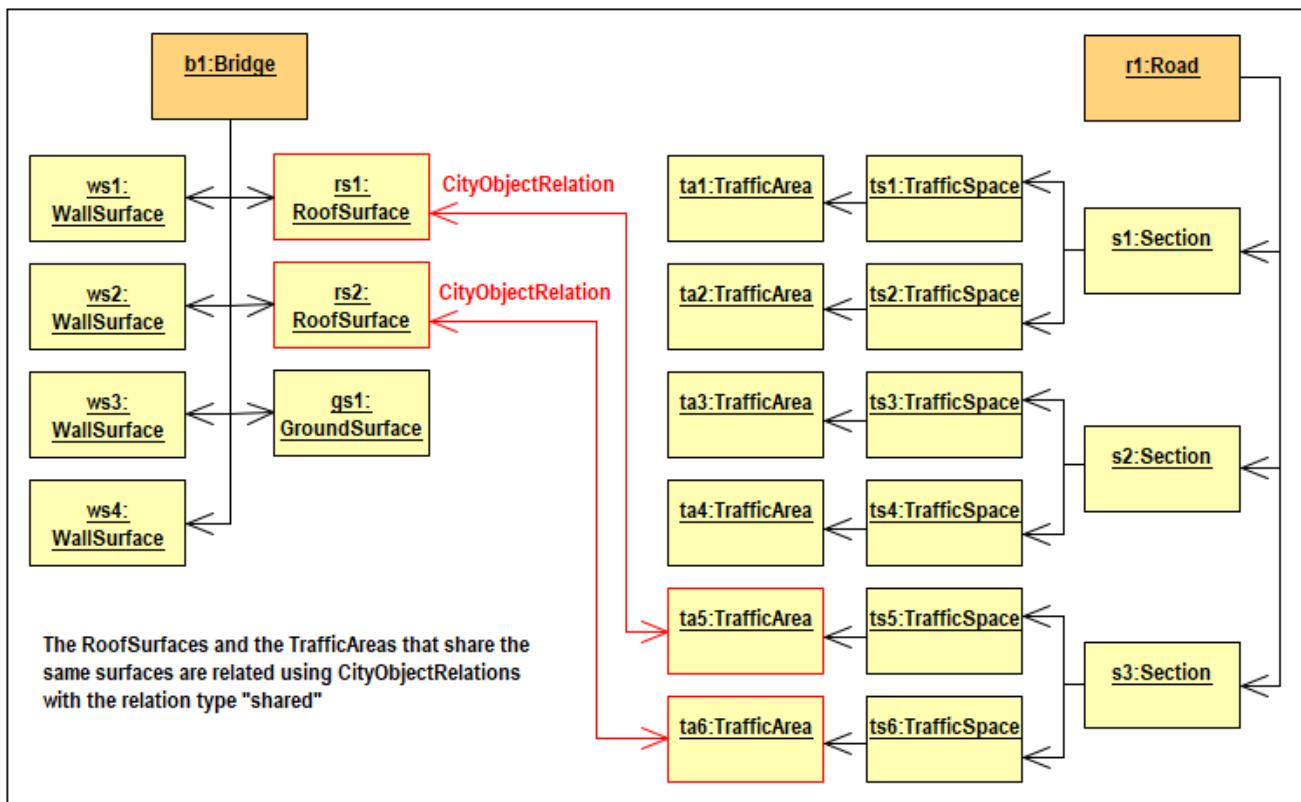


Figure 10. UML object diagram for the building in [Figure 9](#).

Example 6: Parking garage

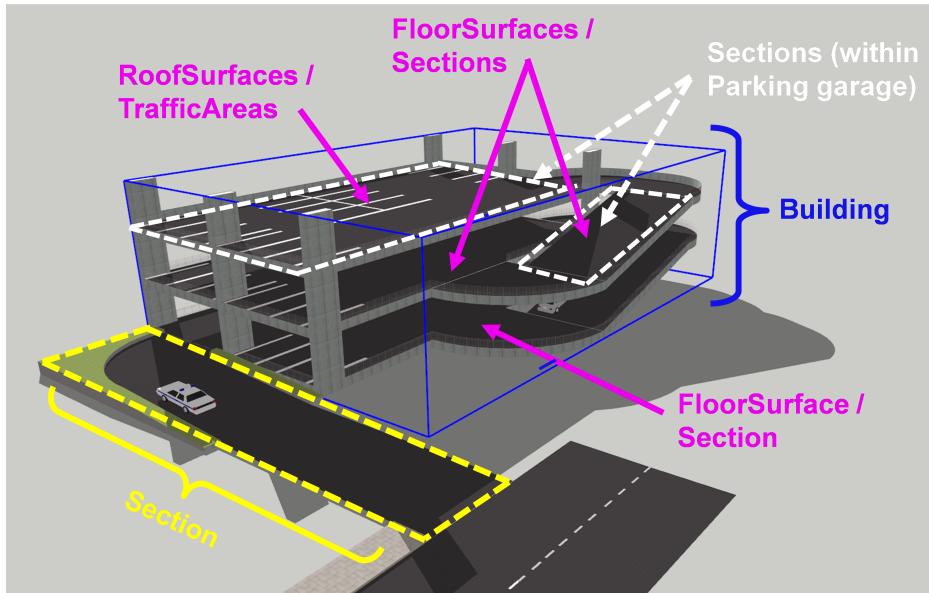


Figure 11. Shared surfaces in a parking garage.

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 on the [CityGML 3.0 GML Encoding GitHub](#).

Example 7: A specific version of a city model

A Version feature groups, for instance, different 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: 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](#).

Referencing features from multiple and different contexts

The same feature can also be used in multiple and different contexts without being able to distinguish between a natural and alternative aggregations. Nevertheless, also in such scenarios it

is recommended to store the feature only once and to reference it from every other context using XLinks in order to avoid duplication and to ensure a non-redundant representation. For this reason, referencing features using XLinks is allowed beyond the scope of Requirement [/req/global/alternativeaggregations](#) even between top-level features. The following examples illustrate some of these scenarios.

Example 9: Building rooms belonging to a Storey

BuildingRooms are usually represented inline as part of the Building they belong to. In addition, Storeys can group BuildingRooms to indicate which BuildingRoom belongs to which Storey. In this case, when the BuildingRooms are already represented inline the Building, the Storeys would reference the BuildingRooms using XLinks to avoid duplication. However, since there is no clear distinction between a natural and alternative aggregations in this example, the BuildingRooms could also be contained inline the corresponding Storey and be referenced from the Building feature.

Example 10: 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. Since there is no natural or alternative aggregation in this example either, one can choose which building part should contain the installation and which building parts should rather use XLink references.

Example 11: Intersection as part of two Roads

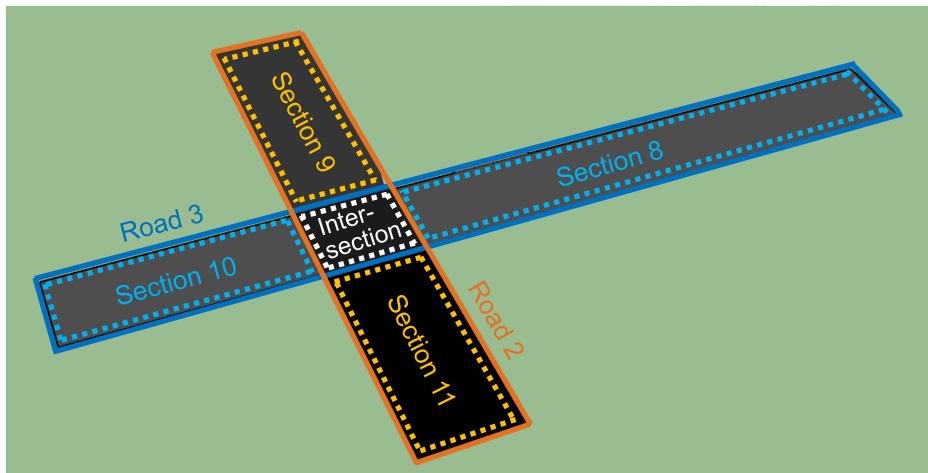


Figure 12. Intersection shared by two roads.

In [Figure 12](#), two Roads (=top-level features) are shown both of which 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 an 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. Also in this example, the roads cannot be semantically distinguished as natural or alternative aggregation of the Intersection.

The GML file is available on the [CityGML 3.0 GML Encoding GitHub](#).

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.

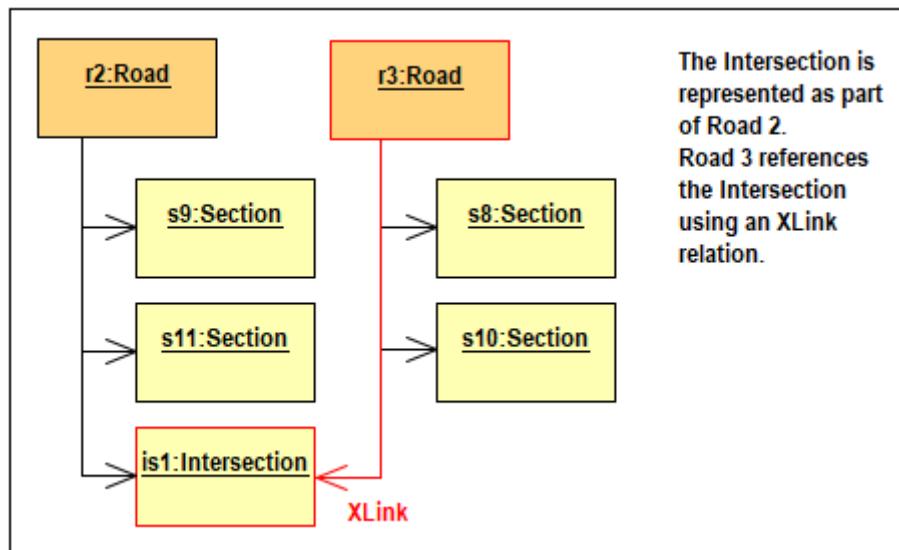


Figure 13. UML object diagram for the building in [Figure 12](#).

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>. A detailed discussion of the Core module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	core.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-core

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

Requirement 7	/req/core/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in core.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractAppearance	«FeatureType» AbstractAppearance
AbstractCityObject	«FeatureType» AbstractCityObject
AbstractDynamizer	«FeatureType» AbstractDynamizer
AbstractFeature	«FeatureType» AbstractFeature
AbstractFeatureWithLifespan	«FeatureType» AbstractFeatureWithLifespan
AbstractLogicalSpace	«FeatureType» AbstractLogicalSpace
AbstractOccupiedSpace	«FeatureType» AbstractOccupiedSpace
AbstractPhysicalSpace	«FeatureType» AbstractPhysicalSpace
AbstractPointCloud	«FeatureType» AbstractPointCloud

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractSpace	«FeatureType» AbstractSpace
AbstractSpaceBoundary	«FeatureType» AbstractSpaceBoundary
AbstractThematicSurface	«FeatureType» AbstractThematicSurface
AbstractUnoccupiedSpace	«FeatureType» AbstractUnoccupiedSpace
AbstractVersion	«FeatureType» AbstractVersion
AbstractVersionTransition	«FeatureType» AbstractVersionTransition
Address	«FeatureType» Address
CityModel	«FeatureType» CityModel
CityObjectRelation	«ObjectType» CityObjectRelation
ClosureSurface	«FeatureType» ClosureSurface
ImplicitGeometry	«ObjectType» ImplicitGeometry
AbstractGenericAttribute	«DataType» AbstractGenericAttribute
ADEOfAbstractAppearance	«DataType» ADEOfAbstractAppearance
ADEOfAbstractCityObject	«DataType» ADEOfAbstractCityObject
ADEOfAbstractDynamizer	«DataType» ADEOfAbstractDynamizer
ADEOfAbstractFeature	«DataType» ADEOfAbstractFeature
ADEOfAbstractFeatureWithLifespan	«DataType» ADEOfAbstractFeatureWithLifespan
ADEOfAbstractLogicalSpace	«DataType» ADEOfAbstractLogicalSpace
ADEOfAbstractOccupiedSpace	«DataType» ADEOfAbstractOccupiedSpace
ADEOfAbstractPhysicalSpace	«DataType» ADEOfAbstractPhysicalSpace
ADEOfAbstractPointCloud	«DataType» ADEOfAbstractPointCloud
ADEOfAbstractSpace	«DataType» ADEOfAbstractSpace
ADEOfAbstractSpaceBoundary	«DataType» ADEOfAbstractSpaceBoundary
ADEOfAbstractThematicSurface	«DataType» ADEOfAbstractThematicSurface
ADEOfAbstractUnoccupiedSpace	«DataType» ADEOfAbstractUnoccupiedSpace
ADEOfAbstractVersion	«DataType» ADEOfAbstractVersion
ADEOfAbstractVersionTransition	«DataType» ADEOfAbstractVersionTransition
ADEOfAddress	«DataType» ADEOfAddress
ADEOfCityModel	«DataType» ADEOfCityModel
ADEOfClosureSurface	«DataType» ADEOfClosureSurface
--- (not mapped to XML, see Section 6.2.2)	«Union» CityModelMember
ExternalReference	«DataType» ExternalReference
Occupancy	«DataType» Occupancy

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
QualifiedArea	«DataType» QualifiedArea
QualifiedVolume	«DataType» QualifiedVolume
--- (not mapped to XML, see Section 6.2.2)	«DataType» XALAddress

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:	
A	If the <i>generalizesTo</i> property (type: <i>gml:ReferenceType</i>) of the <i>AbstractCityObject</i> element is not null, it SHALL contain an XLink reference to an <i>AbstractCityObject</i> element.
B	If the <i>relatedTo</i> property (type: <i>gml:ReferenceType</i>) of the <i>CityObjectRelation</i> is not null, it SHALL contain an XLink reference to an <i>AbstractCityObject</i> element.

Surface boundaries are constrained by the following requirement:

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

[Table 5](#) lists the surfaces that are allowed as space boundaries of the space elements defined in the Core module:

Table 5. Core space elements and their allowed space boundaries

Space element	Allowed space boundaries
<i>core:AbstractLogicalSpace</i>	<ul style="list-style-type: none"> • <i>core:AbstractSpaceBoundary</i> and the substitutable elements: <i>core:AbstractThematicSurface</i>, <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>core:AbstractOccupiedSpace</i>	<ul style="list-style-type: none"> • <i>core:AbstractSpaceBoundary</i> and the substitutable elements: <i>core:AbstractThematicSurface</i>, <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>core:AbstractPhysicalSpace</i>	<ul style="list-style-type: none"> • <i>core:AbstractSpaceBoundary</i> and the substitutable elements: <i>core:AbstractThematicSurface</i>, <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>core:AbstractSpace</i>	<ul style="list-style-type: none"> • <i>core:AbstractSpaceBoundary</i> and the substitutable elements: <i>core:AbstractThematicSurface</i>, <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>core:AbstractUnoccupiedSpace</i>	<ul style="list-style-type: none"> • <i>core:AbstractSpaceBoundary</i> and the substitutable elements: <i>core:AbstractThematicSurface</i>, <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

Recommendation 1	<i>/rec/core/crs</i>
It is <i>strongly recommended</i> 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 <i>srsName</i> attribute on the <i>gml:Envelope</i> which is the value of the <i>gml:boundedBy</i> property of the root <i>core:CityModel</i> 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 <i>srsName</i> .	

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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc50>. A detailed discussion of the Appearance module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-appearance>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	appearance.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-appearance
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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

Requirement 10	/req/appearance/elements
-----------------------	--------------------------

CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in [appearance.xsd](#).

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractSurfaceData	«FeatureType» AbstractSurfaceData
AbstractTexture	«FeatureType» AbstractTexture
Appearance	«FeatureType» Appearance
GeoreferencedTexture	«FeatureType» GeoreferencedTexture
ParameterizedTexture	«FeatureType» ParameterizedTexture
TextureAssociation	«ObjectType» TextureAssociation
X3DMaterial	«FeatureType» X3DMaterial
AbstractTextureParameterization	«DataType» AbstractTextureParameterization
ADEOfAbstractSurfaceData	«DataType» ADEOfAbstractSurfaceData
ADEOfAbstractTexture	«DataType» ADEOfAbstractTexture
ADEOfAppearance	«DataType» ADEOfAppearance
ADEOfGeoreferencedTexture	«DataType» ADEOfGeoreferencedTexture
ADEOfParameterizedTexture	«DataType» ADEOfParameterizedTexture
ADEOfX3DMaterial	«DataType» ADEOfX3DMaterial
TexCoordGen	«DataType» TexCoordGen
TexCoordList	«DataType» TexCoordList

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 <i>textureCoordinates</i> property of the <i>TexCoordList</i> 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 <i>textureCoordinates</i> and <i>ring</i> properties of a <i>TexCoordList</i> element form pairs and their order is decisive. The first <i>textureCoordinates</i> property in the sequence forms a pair with the first <i>ring</i> property in the sequence, the second <i>textureCoordinates</i> property forms a pair with the second <i>ring</i> property, and so on. As a consequence, the number of <i>textureCoordinates</i> and <i>ring</i> properties SHALL be identical.
C	A <i>TexCoordList</i> element SHALL provide <i>textureCoordinates</i> for all <i>gml:LinearRing</i> elements contained in the surface geometry that is referenced by the <i>target</i> property of the embracing <i>TextureAssociation</i> . This explicitly includes both exterior and interior rings.
D	The <i>ring</i> property (type: <i>anyURI</i>) SHALL reference the <i>gml:id</i> of the target <i>gml:LinearRing</i> using an appropriate XPointer.
E	Each point in a ring of a surface geometry SHALL receive a point in texture space. The number of 2D points in the <i>textureCoordinates</i> element therefore SHALL be identical with the number of 3D points in the ring referenced by the corresponding <i>ring</i> property. This explicitly includes texture coordinates for the last point in a <i>gml:LinearRing</i> element which, by GML definition, must be coincident with the first point.
F	The order of points in the <i>textureCoordinates</i> SHALL follow the order of the points in the referenced <i>ring</i> 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 /req/appearance/georeferencedtexture	
Using GeoreferencedTexture elements is subject to the following restrictions:	
A	A GeoreferencedTexture element SHALL define the geo-reference either inline using the <i>referencePoint</i> and <i>orientation</i> properties or externally inside the texture image (e.g., by using the GeoTIFF image format) or through an accompanying world file.
B	The <i>referencePoint</i> property (type: <i>gml:PointPropertyType</i>) SHALL only contain or reference a 2D point geometry.

Recommendation 2 /rec/appearance/global	
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 <i>appearanceMember</i> of a <i>CityModel</i>).	

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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc63>. A detailed discussion of the Bridge module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-bridge>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	bridge.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-bridge
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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

Requirement 14	/req/bridge/elements
----------------	----------------------

CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in [bridge.xsd](#).

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
AbstractBridge	«FeatureType» AbstractBridge
Bridge	«TopLevelFeatureType» Bridge
BridgeConstructiveElement	«FeatureType» BridgeConstructiveElement
BridgeFurniture	«FeatureType» BridgeFurniture
BridgeInstallation	«FeatureType» BridgeInstallation
BridgePart	«FeatureType» BridgePart
BridgeRoom	«FeatureType» BridgeRoom
ADEOfAbstractBridge	«DataType» ADEOfAbstractBridge
ADEOfBridge	«DataType» ADEOfBridge
ADEOfBridgeConstructiveElement	«DataType» ADEOfBridgeConstructiveElement
ADEOfBridgeFurniture	«DataType» ADEOfBridgeFurniture
ADEOfBridgeInstallation	«DataType» ADEOfBridgeInstallation
ADEOfBridgePart	«DataType» ADEOfBridgePart
ADEOfBridgeRoom	«DataType» ADEOfBridgeRoom

Surface boundaries are constrained by the following requirement:

Requirement 15	/req/bridge/boundaries
	<p>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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each 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.</p>

[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

Space element	Allowed thematic surface boundaries
----------------------	--

<i>brid:AbstractBridge</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>brid:Bridge</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>brid:BridgeConstructiveElement</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>brid:BridgeFurniture</i>	No boundaries allowed

<i>brid:BridgeInstallation</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>brid:BridgePart</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>brid:BridgeRoom</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • 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>. A detailed discussion of the Building module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-building>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
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

Requirement 16 /req/building/elements

CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in [building.xsd](#).

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 9. Building XML Elements

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractBuilding	«FeatureType» AbstractBuilding
AbstractBuildingSubdivision	«FeatureType» AbstractBuildingSubdivision
Building	«TopLevelFeatureType» Building
BuildingConstructiveElement	«FeatureType» BuildingConstructiveElement
BuildingFurniture	«FeatureType» BuildingFurniture
BuildingInstallation	«FeatureType» BuildingInstallation
BuildingPart	«FeatureType» BuildingPart
BuildingRoom	«FeatureType» BuildingRoom
BuildingUnit	«FeatureType» BuildingUnit
Storey	«FeatureType» Storey
ADEOfAbstractBuilding	«DataType» ADEOfAbstractBuilding
ADEOfAbstractBuildingSubdivision	«DataType» ADEOfAbstractBuildingSubdivision
ADEOfBuilding	«DataType» ADEOfBuilding
ADEOfBuildingConstructiveElement	«DataType» ADEOfBuildingConstructiveElement
ADEOfBuildingFurniture	«DataType» ADEOfBuildingFurniture
ADEOfBuildingInstallation	«DataType» ADEOfBuildingInstallation
ADEOfBuildingPart	«DataType» ADEOfBuildingPart
ADEOfBuildingRoom	«DataType» ADEOfBuildingRoom
ADEOfBuildingUnit	«DataType» ADEOfBuildingUnit
ADEOfStorey	«DataType» ADEOfStorey
RoomHeight	«DataType» RoomHeight

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*), each 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

Space element	Allowed thematic surface boundaries
<i>bldg:AbstractBuilding</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <ul style="list-style-type: none"> <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>bldg:AbstractBuildingSubdivision</i>	No boundaries allowed
<i>bldg:bldg:Building</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <ul style="list-style-type: none"> <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>bldg:BuildingConstructiveElement</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>bldg:BuildingFurniture</i>	No boundaries allowed
<i>bldg:BuildingInstallation</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>bldg:BuildingPart</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>bldg:BuildingRoom</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <ul style="list-style-type: none"> <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>bldg:BuildingUnit</i>	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>bldg:Storey</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <ul style="list-style-type: none"> <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • 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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc51>. A detailed discussion of the CityFurniture module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-cityfurniture>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	cityFurniture.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityfurniture
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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

Requirement 18	/req/cityfurniture/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in cityFurniture.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
CityFurniture	«TopLevelFeatureType» CityFurniture

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
ADEOfCityFurniture	«DataType» ADEOfCityFurniture

Surface boundaries are constrained by the following requirement:

Requirement 19	/req/cityfurniture/boundaries
<p>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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each 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.</p>	

[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

Space element	Allowed thematic surface boundaries
<i>frn:CityFurniture</i>	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • 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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc52>. A detailed discussion of the CityObjectGroup module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-cityobjectgroup	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	cityObjectGroup.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-cityobjectgroup
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

The CityObjectGroup Conformance Class implements the following Requirements Class from the

CityGML 3.0 Conceptual Model Standard:

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

The applicable GML schema is [cityObjectGroup.xsd](#).

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

Requirement 20	/req/cityobjectgroup/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in cityObjectGroup.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
CityObjectGroup	«TopLevelFeatureType» CityObjectGroup
Role	«ObjectType» Role
ADEOfCityObjectGroup	«DataType» ADEOfCityObjectGroup

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

Surface boundaries are constrained by the following requirement:

Requirement 22	/req/cityobjectgroup/boundaries
<p>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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each 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.</p>	

[Table 14](#) lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the CityObjectGroup module:

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

Space element	Allowed thematic surface boundaries
<i>grp:CityObjectGroup</i>	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc62>. A detailed discussion of the Construction module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-construction>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	construction.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-construction
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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

Requirement 23	/req/construction/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in construction.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractConstruction	«FeatureType» AbstractConstruction
AbstractConstructionSurface	«FeatureType» AbstractConstructionSurface
AbstractConstructiveElement	«FeatureType» AbstractConstructiveElement
AbstractFillingElement	«FeatureType» AbstractFillingElement
AbstractFillingSurface	«FeatureType» AbstractFillingSurface
AbstractFurniture	«FeatureType» AbstractFurniture
AbstractInstallation	«FeatureType» AbstractInstallation
CeilingSurface	«FeatureType» CeilingSurface
Door	«FeatureType» Door
DoorSurface	«FeatureType» DoorSurface
FloorSurface	«FeatureType» FloorSurface
GroundSurface	«FeatureType» GroundSurface
InteriorWallSurface	«FeatureType» InteriorWallSurface
OtherConstruction	«TopLevelFeatureType» OtherConstruction
OuterCeilingSurface	«FeatureType» OuterCeilingSurface
OuterFloorSurface	«FeatureType» OuterFloorSurface
RoofSurface	«FeatureType» RoofSurface
WallSurface	«FeatureType» WallSurface
Window	«FeatureType» Window
WindowSurface	«FeatureType» WindowSurface
ADEOfAbstractConstruction	«DataType» ADEOfAbstractConstruction
ADEOfAbstractConstructionSurface	«DataType» ADEOfAbstractConstructionSurface
ADEOfAbstractConstructiveElement	«DataType» ADEOfAbstractConstructiveElement
ADEOfAbstractFillingElement	«DataType» ADEOfAbstractFillingElement
ADEOfAbstractFillingSurface	«DataType» ADEOfAbstractFillingSurface
ADEOfAbstractFurniture	«DataType» ADEOfAbstractFurniture
ADEOfAbstractInstallation	«DataType» ADEOfAbstractInstallation
ADEOfCeilingSurface	«DataType» ADEOfCeilingSurface
ADEOfDoor	«DataType» ADEOfDoor
ADEOfDoorSurface	«DataType» ADEOfDoorSurface
ADEOfFloorSurface	«DataType» ADEOfFloorSurface
ADEOfGroundSurface	«DataType» ADEOfGroundSurface
ADEOfInteriorWallSurface	«DataType» ADEOfInteriorWallSurface

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
ADEOfOtherConstruction	«DataType» ADEOfOtherConstruction
ADEOfOuterCeilingSurface	«DataType» ADEOfOuterCeilingSurface
ADEOfOuterFloorSurface	«DataType» ADEOfOuterFloorSurface
ADEOfRoofSurface	«DataType» ADEOfRoofSurface
ADEOfWallSurface	«DataType» ADEOfWallSurface
ADEOfWindow	«DataType» ADEOfWindow
ADEOfWindowSurface	«DataType» ADEOfWindowSurface
ConstructionEvent	«DataType» ConstructionEvent
Elevation	«DataType» Elevation
Height	«DataType» Height

Surface boundaries are constrained by the following requirement:

Requirement 24	/req/construction/boundaries
<p>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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each 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.</p>	

[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

Space element	Allowed thematic surface boundaries
<i>con:AbstractConstruction</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>con:AbstractConstructiveElement</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>con:AbstractFillingElement</i>	No boundaries allowed
<i>con:AbstractFurniture</i>	No boundaries allowed
<i>con:AbstractInstallation</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>con:Door</i>	<ul style="list-style-type: none"> • <i>con:DoorSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>con:OtherConstruction</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>con:Window</i>	<ul style="list-style-type: none"> • <i>con:WindowSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • 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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc53>. A detailed discussion of the Dynamizer module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-dynamizer>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	dynamizer.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-dynamizer
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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

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

The applicable GML schema is [dynamizer.xsd](#).

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 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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractAtomicTimeseries	«FeatureType» AbstractAtomicTimeseries
AbstractTimeseries	«FeatureType» AbstractTimeseries
CompositeTimeseries	«FeatureType» CompositeTimeseries
Dynamizer	«FeatureType» Dynamizer
GenericTimeseries	«FeatureType» GenericTimeseries
StandardFileTimeseries	«FeatureType» StandardFileTimeseries

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
TabulatedFileTimeseries	«FeatureType» TabulatedFileTimeseries
ADEOfAbstractAtomicTimeseries	«DataType» ADEOfAbstractAtomicTimeseries
ADEOfAbstractTimeseries	«DataType» ADEOfAbstractTimeseries
ADEOfCompositeTimeseries	«DataType» ADEOfCompositeTimeseries
ADEOfDynamizer	«DataType» ADEOfDynamizer
ADEOfGenericTimeseries	«DataType» ADEOfGenericTimeseries
ADEOfStandardFileTimeseries	«DataType» ADEOfStandardFileTimeseries
ADEOfTabulatedFileTimeseries	«DataType» ADEOfTabulatedFileTimeseries
SensorConnection	«DataType» SensorConnection
TimeseriesComponent	«DataType» TimeseriesComponent
TimeValuePair	«DataType» TimeValuePair

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:

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

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 City GML 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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc54>. A detailed discussion of the Generics module can also be found in the [CityGML](#)

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-generics	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	generics.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-generics
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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 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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
GenericLogicalSpace	«TopLevelFeatureType» GenericLogicalSpace
GenericOccupiedSpace	«TopLevelFeatureType» GenericOccupiedSpace
GenericThematicSurface	«TopLevelFeatureType» GenericThematicSurface
GenericUnoccupiedSpace	«TopLevelFeatureType» GenericUnoccupiedSpace
ADEOfGenericLogicalSpace	«DataType» ADEOfGenericLogicalSpace
ADEOfGenericOccupiedSpace	«DataType» ADEOfGenericOccupiedSpace
ADEOfGenericThematicSurface	«DataType» ADEOfGenericThematicSurface
ADEOfGenericUnoccupiedSpace	«DataType» ADEOfGenericUnoccupiedSpace
CodeAttribute	«DataType» CodeAttribute
DateAttribute	«DataType» DateAttribute
DoubleAttribute	«DataType» DoubleAttribute
GenericAttributeSet	«DataType» GenericAttributeSet
IntAttribute	«DataType» IntAttribute
MeasureAttribute	«DataType» MeasureAttribute
StringAttribute	«DataType» StringAttribute
UriAttribute	«DataType» UriAttribute

Surface boundaries are constrained by the following requirement:

Requirement 29	/req/generics/boundaries
<p>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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each 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.</p>	

[Table 19](#) lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the Generics module:

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

Space element	Allowed thematic surface boundaries
gen:GenericLogicalSpace	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • gen:GenericThematicSurface • possible surface elements from ADEs

<i>gen:GenericOccupiedSpace</i>	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>gen:GenericUnoccupiedSpace</i>	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

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>. A detailed discussion of the LandUse module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-landuse>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	landUse.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-landuse
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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

Requirement 30	/req/landuse/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in landUse.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
LandUse	«TopLevelFeatureType» LandUse
ADEOfLandUse	«DataType» ADEOfLandUse

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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc56>. A detailed discussion of the PointCloud module can also be found in the CityGML 3.0 Users Guide.

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-pointcloud	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	pointCloud.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-pointcloud
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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

Requirement 31	/req/PointCloud/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in pointCloud.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
PointCloud	«FeatureType» PointCloud
ADEOfPointCloud	«DataType» ADEOfPointCloud

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>. A detailed discussion of the Relief module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-relief>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	relief.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-relief
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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 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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
AbstractReliefComponent	«FeatureType» AbstractReliefComponent
BreaklineRelief	«FeatureType» BreaklineRelief
MassPointRelief	«FeatureType» MassPointRelief
RasterRelief	«FeatureType» RasterRelief
ReliefFeature	«TopLevelFeatureType» ReliefFeature
TINRelief	«FeatureType» TINRelief
ADEOfAbstractReliefComponent	«DataType» ADEOfAbstractReliefComponent
ADEOfBreaklineRelief	«DataType» ADEOfBreaklineRelief
ADEOfMassPointRelief	«DataType» ADEOfMassPointRelief
ADEOfRasterRelief	«DataType» ADEOfRasterRelief
ADEOfReliefFeature	«DataType» ADEOfReliefFeature
ADEOfTINRelief	«DataType» ADEOfTINRelief

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>. A detailed discussion of the Transportation module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

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

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	transportation.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-transportation

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

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

Requirement 33	/req/transportation/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in transportation.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractTransportationSpace	«FeatureType» AbstractTransportationSpace

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
AuxiliaryTrafficArea	«FeatureType» AuxiliaryTrafficArea
AuxiliaryTrafficSpace	«FeatureType» AuxiliaryTrafficSpace
ClearanceSpace	«FeatureType» ClearanceSpace
Hole	«FeatureType» Hole
HoleSurface	«FeatureType» HoleSurface
Intersection	«FeatureType» Intersection
Marking	«FeatureType» Marking
Railway	«TopLevelFeatureType» Railway
Road	«TopLevelFeatureType» Road
Section	«FeatureType» Section
Square	«TopLevelFeatureType» Square
Track	«TopLevelFeatureType» Track
TrafficArea	«FeatureType» TrafficArea
TrafficSpace	«FeatureType» TrafficSpace
Waterway	«TopLevelFeatureType» Waterway
ADEOfAbstractTransportationSpace	«DataType» ADEOfAbstractTransportationSpace
ADEOfAuxiliaryTrafficArea	«DataType» ADEOfAuxiliaryTrafficArea
ADEOfAuxiliaryTrafficSpace	«DataType» ADEOfAuxiliaryTrafficSpace
ADEOfClearanceSpace	«DataType» ADEOfClearanceSpace
ADEOfHole	«DataType» ADEOfHole
ADEOfHoleSurface	«DataType» ADEOfHoleSurface
ADEOfIntersection	«DataType» ADEOfIntersection
ADEOfMarking	«DataType» ADEOfMarking
ADEOfRailway	«DataType» ADEOfRailway
ADEOfRoad	«DataType» ADEOfRoad
ADEOfSection	«DataType» ADEOfSection
ADEOfSquare	«DataType» ADEOfSquare
ADEOfTrack	«DataType» ADEOfTrack
ADEOfTrafficArea	«DataType» ADEOfTrafficArea
ADEOfTrafficSpace	«DataType» ADEOfTrafficSpace
ADEOfWaterway	«DataType» ADEOfWaterway

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

Surface boundaries are constrained by the following requirement:

Requirement 35	/req/transportation/boundaries
<p>Table 24 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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property SHALL contain exactly one surface element from Table 24 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.</p>	

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

Space element	Allowed thematic surface boundaries
<i>tran:AbstractTransportationSpace</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:AuxiliaryTrafficSpace</i>	<ul style="list-style-type: none"> • <i>tran:AuxiliaryTrafficArea</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:ClearanceSpace</i>	<ul style="list-style-type: none"> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>tran:Hole</i>	<ul style="list-style-type: none"> • <i>tran:HoleSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Intersection</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Railway</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Road</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Section</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Square</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Track</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>tran:TrafficSpace</i>	<ul style="list-style-type: none"> • <i>tran:TrafficArea</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tran:Waterway</i>	<ul style="list-style-type: none"> • <i>tran:Marking</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • 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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc65>. A detailed discussion of the Tunnel module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-tunnel	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	tunnel.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-tunnel
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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

Requirement 36	/req/tunnel/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in tunnel.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractTunnel	«FeatureType» AbstractTunnel
HollowSpace	«FeatureType» HollowSpace
Tunnel	«TopLevelFeatureType» Tunnel
TunnelConstructiveElement	«FeatureType» TunnelConstructiveElement
TunnelFurniture	«FeatureType» TunnelFurniture
TunnelInstallation	«FeatureType» TunnelInstallation
TunnelPart	«FeatureType» TunnelPart
ADEOfAbstractTunnel	«DataType» ADEOfAbstractTunnel
ADEOfHollowSpace	«DataType» ADEOfHollowSpace
ADEOfTunnel	«DataType» ADEOfTunnel
ADEOfTunnelConstructiveElement	«DataType» ADEOfTunnelConstructiveElement
ADEOfTunnelFurniture	«DataType» ADEOfTunnelFurniture
ADEOfTunnelInstallation	«DataType» ADEOfTunnelInstallation
ADEOfTunnelPart	«DataType» ADEOfTunnelPart

Surface boundaries are constrained by the following requirement:

Requirement 37	/req/tunnel/boundaries

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*), each 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

Space element	Allowed thematic surface boundaries
<i>tun:AbstractTunnel</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <ul style="list-style-type: none"> <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tun:HollowSpace</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <ul style="list-style-type: none"> <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>tun:Tunnel</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tun:TunnelConstructiveElement</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs
<i>tun:TunnelFurniture</i>	No boundaries allowed
<i>tun:TunnelInstallation</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

<i>tun:TunnelPart</i>	<ul style="list-style-type: none"> • <i>con:AbstractConstructionSurface</i> and the substitutable elements: <i>con:GroundSurface</i>, <i>con:RoofSurface</i>, <i>con:CeilingSurface</i>, <i>con:OuterCeilingSurface</i>, <i>con:FloorSurface</i>, <i>con:OuterFloorSurface</i>, <i>con:WallSurface</i>, <i>con:InteriorWallSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • 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>. A detailed discussion of the Vegetation module can also be found in the [CityGML 3.0 Users Guide](#).

Requirements Class

<http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-vegetation>

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	vegetation.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-vegetation
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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

Requirement 38	/req/vegetation/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in vegetation.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model; Data Dictionary)
AbstractVegetationObject	«FeatureType» AbstractVegetationObject
PlantCover	«TopLevelFeatureType» PlantCover
SolitaryVegetationObject	«TopLevelFeatureType» SolitaryVegetationObject
ADEOfAbstractVegetationObject	«DataType» ADEOfAbstractVegetationObject
ADEOfPlantCover	«DataType» ADEOfPlantCover
ADEOfSolitaryVegetationObject	«DataType» ADEOfSolitaryVegetationObject

Surface boundaries are constrained by the following requirement:

Requirement 39	/req/vegetation/boundaries
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 <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each 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.	

[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

Space element	Allowed thematic surface boundaries
<i>veg:AbstractVegetationObject</i>	No boundaries allowed
<i>veg:PlantCover</i>	No boundaries allowed
<i>veg:SolitaryVegetationObject</i>	No boundaries allowed

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>. A detailed discussion of the Versioning module can also be found in the CityGML 3.0 Users Guide.

Requirements Class	
http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-versioning	
Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	versioning.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-versioning
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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

Requirement 40	/req/versioning/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in versioning.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
Version	«FeatureType» Version
VersionTransition	«FeatureType» VersionTransition
ADEOfVersion	«DataType» ADEOfVersion
ADEOfVersionTransition	«DataType» ADEOfVersionTransition
Transaction	«DataType» Transaction

Requirement 41	/req/versioning/reference

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

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

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>, the Data Dictionary is available under <https://docs.ogc.org/is/20-010/20-010.html#toc61>. A detailed discussion of the WaterBody module can also be found in the CityGML 3.0 Users Guide.

Requirements Class

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

Target type	Implementations of the CityGML 3.0 Conceptual Model using GML encodings
Dependency	waterBody.xsd
Dependency	http://www.opengis.net/spec/CityGML-1/3.0/req/req-class-waterbody
Dependency	http://www.opengis.net/spec/CityGML-2/3.0/req/req-class-core

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>

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

Requirement 42	/req/waterbody/elements
CityGML XML elements implemented by a conforming instance document shall conform to the XML schema in waterBody.xsd .	

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

XML Schema Element (XML Schema documentation)	Conceptual Model (UML model ; Data Dictionary)
AbstractWaterBoundarySurface	«FeatureType» AbstractWaterBoundarySurface
WaterBody	«TopLevelFeatureType» WaterBody
WaterGroundSurface	«FeatureType» WaterGroundSurface
WaterSurface	«FeatureType» WaterSurface
ADEOfAbstractWaterBoundarySurface	«DataType» ADEOfAbstractWaterBoundarySurface
ADEOfWaterBody	«DataType» ADEOfWaterBody
ADEOfWaterGroundSurface	«DataType» ADEOfWaterGroundSurface
ADEOfWaterSurface	«DataType» ADEOfWaterSurface

Surface boundaries are constrained by the following requirement:

Requirement 43	/req/waterbody/boundaries

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*), each 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 31 lists the surfaces that are allowed as thematic surface boundaries of the space elements defined in the WaterBody module:

Table 31. WaterBody space elements and their allowed thematic surface boundaries

Space element	Allowed thematic surface boundaries
<i>wtr:WaterBody</i>	<ul style="list-style-type: none"> • <i>wtr:AbstractWaterBoundarySurface</i> and the substitutable elements: <i>wtr:WaterGroundSurface</i>, <i>wtr:WaterSurface</i> • <i>core:ClosureSurface</i> • <i>gen:GenericThematicSurface</i> • possible surface elements from ADEs

Chapter 7. Media Types

Recommendation 3 /rec/mediatype/gml

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)

A.1. Conformance Class Global

Abstract Test 1	
Test ID:	/conf/global/gml
Requirement:	/req/global/gml
Test purpose:	Verify that XML instance documents claiming conformance to this specification validate against the XML schema files.
Test method:	Validate the instance document against the XML schemas to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 2	
Test ID:	/conf/global/referencinggeometries1
Requirement:	/req/global/referencinggeometries1
Test purpose:	Verify that XLinks are not used to reference geometries, except for geometries of <i>ImplicitGeometry</i> elements, from another top-level feature.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 3	
Test ID:	/conf/global/referencinggeometries2
Requirement:	/req/global/referencinggeometries2
Test purpose:	When referencing geometries of spaces and space boundaries, verify that: <ul style="list-style-type: none">• Geometries stored inline a space boundary are not redundantly stored as geometry of a space. If the geometry is stored with the space in addition, it references the geometry from the space boundary using XLinks.• Space boundaries do not reference geometries of a space using XLinks.• This applies for all spaces and space boundaries that are children of the same top-level feature.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 4	
Test ID:	/conf/global/referencinggeometries3
Requirement:	/req/global/referencinggeometries3

Test purpose:	Verify that LoDs are self-contained: Geometries are not shared between different LoDs using XLinks.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 5

Test ID:	/conf/global/referencinggeometries4
Requirement:	/req/global/referencinggeometries4
Test purpose:	<p>When referencing shared geometries between top-level features using <i>CityObjectRelation</i> elements, verify that:</p> <ul style="list-style-type: none"> If two top-level features share a common geometry, the shared geometry is stored for each top-level feature separately. If <i>CityObjectRelation</i> elements are used to denote that a geometry is shared between two top-level features, then a <i>CityObjectRelation</i> element is stored for each feature (might be the top-level feature itself or one of its nested features), and each <i>CityObjectRelation</i> element is assigned the relation type “shared” and references the other feature using an XLink. Thus, the reference is bi-directional.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 6

Test ID:	/conf/global/alternativeaggregations
Requirement:	/req/global/alternativeaggregations
Test purpose:	<p>When referencing features from alternative aggregations, verify that:</p> <ul style="list-style-type: none"> Each feature belongs to a natural aggregation hierarchy and is stored inline this hierarchy. Alternative aggregations do not contain the feature inline but use an XLink to reference the feature.
Test method:	Inspect the instance document to verify the above requirement.

A.2. Conformance Class Core

Abstract Test 7

Test ID:	/conf/core/elements
Requirement:	/req/core/elements
Test purpose:	Verify that instance documents using the Core XML elements listed in Table 4 validate against the XML schema specified in http://schemas.opengis.net/citygml/3.0/core.xsd .

Test method:	Validate the instance document against the core.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.
---------------------	--

Abstract Test 8

Test ID:	/conf/core/reference
Requirement:	/req/core/reference
Test purpose:	<p>For the following properties, verify that:</p> <ul style="list-style-type: none"> • If the <i>generalizesTo</i> property (type: <i>gml:ReferenceType</i>) of the <i>AbstractCityObject</i> element is not null, it contains an XLink reference to an <i>AbstractCityObject</i> element. • If the <i>relatedTo</i> property (type: <i>gml:ReferenceType</i>) of the <i>CityObjectRelation</i> element is not null, it contains an XLink reference to an <i>AbstractCityObject</i> element.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 9

Test ID:	/conf/core/boundaries
Requirement:	/req/core/boundaries
Test purpose:	<p>For each Core space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 5 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.</p>
Test method:	Inspect the instance document to verify the above requirement.

A.3. Conformance Class Appearance

Abstract Test 10

Test ID:	/conf/appearance/elements
Requirement:	/req/appearance/elements
Test purpose:	<p>Verify that instance documents using the Appearance XML elements listed in Table 6 validate against the XML schema specified in http://schemas.opengis.net/citygml/appearance/3.0/appearance.xsd.</p>
Test method:	Validate the instance document against the appearance.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 11

Test ID:	/conf/appearance/target
Requirement:	/req/appearance/target
Test purpose:	Verify that surface data is only applied to surface geometries. The <i>target</i> property of a surface data element therefore only references a subtype of <i>gml:AbstractSurfaceType</i> or a <i>gml:MultiSurface</i> .
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 12

Test ID:	/conf/appearance/parameterizedtexture
Requirement:	/req/appearance/parameterizedtexture
Test purpose:	<p>When assigning texture coordinates to a surface geometry using <i>ParameterizedTexture</i> elements, verify that:</p> <ul style="list-style-type: none">• Texture coordinates given by the <i>textureCoordinates</i> property of the <i>TexCoordList</i> element define an explicit mapping of a surface's boundary points to points in texture space. A point in texture space is given as a coordinate pair consisting of two doubles.• The <i>textureCoordinates</i> and <i>ring</i> properties of a <i>TexCoordList</i> element form pairs and their order is decisive. The first <i>textureCoordinates</i> property in the sequence forms a pair with the first <i>ring</i> property in the sequence, the second <i>textureCoordinates</i> property forms a pair with the second <i>ring</i> property, and so on. As a consequence, the number of <i>textureCoordinates</i> and <i>ring</i> properties is identical.• A <i>TexCoordList</i> element provides <i>textureCoordinates</i> for all <i>gml:LinearRing</i> elements contained in the surface geometry that is referenced by the <i>target</i> property of the embracing <i>TextureAssociation</i>. This explicitly includes both exterior and interior rings.• The <i>ring</i> property (type: <i>anyURI</i>) references the <i>gml:id</i> of the target <i>gml:LinearRing</i> using an appropriate XPointer.• Each point in a ring of a surface geometry receives a point in texture space. The number of 2D points in the <i>textureCoordinates</i> element therefore is identical with the number of 3D points in the ring referenced by the corresponding <i>ring</i> property. This explicitly includes texture coordinates for the last point in a <i>gml:LinearRing</i> element which, by GML definition, must be coincident with the first point.• The order of points in the <i>textureCoordinates</i> follows the order of the points in the referenced <i>ring</i> element as given in the CityGML document regardless of a possibly flipped surface orientation.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 13	
Test ID:	/conf/appearance/georeferencedtexture
Requirement:	/req/appearance/georeferencedtexture
Test purpose:	<p>When using <i>GeoreferencedTexture</i> elements, verify that:</p> <ul style="list-style-type: none"> • A <i>GeoreferencedTexture</i> element defines the geo-reference either inline using the <i>referencePoint</i> and <i>orientation</i> properties or externally inside the texture image (e.g., by using the GeoTIFF image format) or through an accompanying world file. • The <i>referencePoint</i> property (type: <i>gml:PointPropertyType</i>) only contains or references a 2D point geometry.
Test method:	Inspect the instance document to verify the above requirement.

A.4. Conformance Class Bridge

Abstract Test 14	
Test ID:	/conf/bridge/elements
Requirement:	/req/bridge/elements
Test purpose:	Verify that instance documents using the Bridge XML elements listed in Table 7 validate against the XML schema specified in http://schemas.opengis.net/citygml/bridge/3.0/bridge.xsd .
Test method:	Validate the instance document against the bridge.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 15	
Test ID:	/conf/bridge/boundaries
Requirement:	/req/bridge/boundaries
Test purpose:	For each Bridge space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 8 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.5. Conformance Class Building

Abstract Test 16	

Test ID:	/conf/building/elements
Requirement:	/req/building/elements
Test purpose:	Verify that instance documents using the Building XML elements listed in Table 9 validate against the XML schema specified in http://schemas.opengis.net/citygml/building/3.0/building.xsd .
Test method:	Validate the instance document against the building.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 17

Test ID:	/conf/building/boundaries
Requirement:	/req/building/boundaries
Test purpose:	For each Building space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 10 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.6. Conformance Class CityFurniture

Abstract Test 18

Test ID:	/conf/cityfurniture/elements
Requirement:	/req/cityfurniture/elements
Test purpose:	Verify that instance documents using the CityFurniture XML elements listed in Table 11 validate against the XML schema specified in http://schemas.opengis.net/citygml/cityfurniture/3.0/cityFurniture.xsd .
Test method:	Validate the instance document against the cityFurniture.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 19

Test ID:	/conf/cityfurniture/boundaries
Requirement:	/req/cityfurniture/boundaries

Test purpose:	For each CityFurniture space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 12 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.7. Conformance Class CityObjectGroup

Abstract Test 20	
Test ID:	/conf/cityobjectgroup/elements
Requirement:	/req/cityobjectgroup/elements
Test purpose:	Verify that instance documents using the CityObjectGroup XML elements listed in Table 13 validate against the XML schema specified in http://schemas.opengis.net/citygml/cityobjectgroup/3.0/cityObjectGroup.xsd .
Test method:	Validate the instance document against the cityObjectGroup.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 21	
Test ID:	/conf/cityobjectgroup/reference
Requirement:	/req/cityobjectgroup/reference
Test purpose:	For the following properties, verify that: <ul style="list-style-type: none"> • If the <i>parent</i> property (type: <i>gml:ReferenceType</i>) of the <i>CityObjectGroup</i> element is not null, it contains an XLink reference to a <i>core:AbstractCityObject</i> element. • If the <i>groupMember</i> property (type: <i>gml:AbstractFeatureMemberType</i>) of the <i>Role</i> element is not null, it contains an XLink reference to a <i>core:AbstractCityObject</i> element.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 22	
Test ID:	/conf/cityobjectgroup/boundaries
Requirement:	/req/cityobjectgroup/boundaries

Test purpose:	For each CityObjectGroup space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 14 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.8. Conformance Class Construction

Abstract Test 23	
Test ID:	/conf/construction/elements
Requirement:	/req/construction/elements
Test purpose:	Verify that instance documents using the Construction XML elements listed in Table 15 validate against the XML schema specified in http://schemas.opengis.net/citygml/construction/3.0/construction.xsd .
Test method:	Validate the instance document against the construction.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 24	
Test ID:	/conf/construction/boundaries
Requirement:	/req/construction/boundaries
Test purpose:	For each Construction space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 16 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.9. Conformance Class Dynamizer

Abstract Test 25	
Test ID:	/conf/dynamizer/elements
Requirement:	/req/dynamizer/elements
Test purpose:	Verify that instance documents using the Dynamizer XML elements listed in Table 17 validate against the XML schema specified in http://schemas.opengis.net/citygml/dynamizer/3.0/dynamizer.xsd .

Test method:	Validate the instance document against the dynamizer.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.
---------------------	---

Abstract Test 26

Test ID:	/conf/dynamizer/reference
Requirement:	/req/dynamizer/reference
Test purpose:	Verify that if the <i>sensorLocation</i> property (type: <i>gml:ReferenceType</i>) of the <i>SensorConnection</i> element is not null, it contains an XLink reference to a <i>core:AbstractCityObject</i> element.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 27

Test ID:	/conf/dynamizer/attributeref
Requirement:	/req/dynamizer/attributeref
Test purpose:	<p>When referencing a feature property value using the <i>attributeRef</i> property of a <i>Dynamizer</i> element, verify that:</p> <ul style="list-style-type: none"> • The <i>attributeRef</i> property (type: <i>xs:string</i>) references a feature property value using XPath. • The feature property referenced by the <i>attributeRef</i> property exists. • The data type of the timeseries data provided by a <i>Dynamizer</i> element is equal to the data type of the static value of the feature property that is referenced by the <i>attributeRef</i> property or is of a type that can be converted to the type of the feature property without information loss.
Test method:	Inspect the instance document to verify the above requirement.

A.10. Conformance Class Generics

Abstract Test 28

Test ID:	/conf/generics/elements
Requirement:	/req/generics/elements
Test purpose:	Verify that instance documents using the Generics XML elements listed in Table 18 validate against the XML schema specified in http://schemas.opengis.net/citygml/generics/3.0/generics.xsd .

Test method: Validate the instance document against the generics.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 29

Test ID:	/conf/generics/boundaries
Requirement:	/req/generics/boundaries
Test purpose:	For each Generics space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 19 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.11. Conformance Class LandUse

Abstract Test 30

Test ID:	/conf/landuse/elements
Requirement:	/req/landuse/elements
Test purpose:	Verify that instance documents using the LandUse XML elements listed in Table 20 validate against the XML schema specified in http://schemas.opengis.net/citygml/landuse/3.0/landUse.xsd .
Test method:	Validate the instance document against the landUse.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.12. Conformance Class PointCloud

Abstract Test 31

Test ID:	/conf/pointcloud/elements
Requirement:	/req/pointcloud/elements
Test purpose:	Verify that instance documents using the PointCloud XML elements listed in Table 21 validate against the XML schema specified in http://schemas.opengis.net/citygml/pointcloud/3.0/pointCloud.xsd .
Test method:	Validate the instance document against the pointCloud.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.13. Conformance Class Relief

Abstract Test 32

Test ID:	/conf/relief/elements
Requirement:	/req/relief/elements
Test purpose:	Verify that instance documents using the Relief XML elements listed in Table 22 validate against the XML schema specified in http://schemas.opengis.net/citygml/relief/3.0/relief.xsd .
Test method:	Validate the instance document against the relief.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

A.14. Conformance Class Transportation

Abstract Test 33	
Test ID:	/conf/transportation/elements
Requirement:	/req/transportation/elements
Test purpose:	Verify that instance documents using the Transportation XML elements listed in Table 23 validate against the XML schema specified in http://schemas.opengis.net/citygml/transportation/3.0/transportation.xsd .
Test method:	Validate the instance document against the transportation.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 34	
Test ID:	/conf/transportation/reference
Requirement:	/req/transportation/reference
Test purpose:	For the following properties, verify that: <ul style="list-style-type: none"> • if the <i>predecessor</i> property (type: <i>gml:ReferenceType</i>) of the <i>TrafficSpace</i> element is not null, it contains an XLink reference to a <i>TrafficSpace</i> element. • if the <i>successor</i> property (type: <i>gml:ReferenceType</i>) of the <i>TrafficSpace</i> element is not null, it contains an XLink reference to a <i>TrafficSpace</i> element.
Test method:	Inspect the instance document to verify the above requirement.

Abstract Test 35	
Test ID:	/conf/transportation/boundaries
Requirement:	/req/transportation/boundaries

Test purpose:	For each Transportation space element verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 24 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.15. Conformance Class Tunnel

Abstract Test 36	
Test ID:	/conf/tunnel/elements
Requirement:	/req/tunnel/elements
Test purpose:	Verify that instance documents using the Tunnel XML elements listed in Table 25 validate against the XML schema specified in http://schemas.opengis.net/citygml/tunnel/3.0/tunnel.xsd .
Test method:	Validate the instance document against the tunnel.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 37	
Test ID:	/conf/tunnel/boundaries
Requirement:	/req/tunnel/boundaries
Test purpose:	For each Tunnel space element, verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 26 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.16. Conformance Class Vegetation

Abstract Test 38	
Test ID:	/conf/vegetation/elements
Requirement:	/req/vegetation/elements
Test purpose:	Verify that instance documents using the Vegetation XML elements listed in Table 27 validate against the XML schema specified in http://schemas.opengis.net/citygml/vegetation/3.0/vegetation.xsd .

Test method:	Validate the instance document against the vegetation.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.
---------------------	--

Abstract Test 39

Test ID:	/conf/vegetation/boundaries
Requirement:	/req/vegetation/boundaries
Test purpose:	For each Vegetation space element, verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 28 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.
Test method:	Inspect the instance document to verify the above requirement.

A.17. Conformance Class Versioning

Abstract Test 40

Test ID:	/conf/versioning/elements
Requirement:	/req/versioning/elements
Test purpose:	Verify that instance documents using the Versioning XML elements listed in Table 29 validate against the XML schema specified in http://schemas.opengis.net/citygml/versioning/3.0/versioning.xsd .
Test method:	Validate the instance document against the versioning.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 41

Test ID:	/conf/versioning/reference
Requirement:	/req/versioning/reference

Test purpose:	For the following properties, verify that:
	<ul style="list-style-type: none"> • If the <i>from</i> property (type: <i>gml:ReferenceType</i>) of the <i>VersionTransition</i> element is not null, it only contains an XLink reference to a <i>Version</i> element. • If the <i>to</i> property (type: <i>gml:ReferenceType</i>) of the <i>VersionTransition</i> element is not null, it only contains an XLink reference to a <i>Version</i> element. • If the <i>oldFeature</i> property (type: <i>gml:ReferenceType</i>) of the <i>Transaction</i> element is not null, it only contains an XLink reference to a <i>core:AbstractFeatureWithLifespan</i> element. • If the <i>newFeature</i> property (type: <i>gml:ReferenceType</i>) of the <i>Transaction</i> element is not null, it only contains an XLink reference to a <i>core:AbstractFeatureWithLifespan</i> element. • If the <i>versionMember</i> property (type: <i>gml:AbstractFeatureMemberType</i>) of the <i>Version</i> element is not null, it only contains an XLink reference to a <i>core:AbstractFeatureWithLifespan</i> element.
Test method:	Inspect the instance document to verify the above requirement.

A.18. Conformance Class WaterBody

Abstract Test 42	
Test ID:	/conf/waterbody/elements
Requirement:	/req/waterbody/elements
Test purpose:	Verify that instance documents using the WaterBody XML elements listed in Table 30 validate against the XML schema specified in http://schemas.opengis.net/citygml/waterbody/3.0/waterBody.xsd .
Test method:	Validate the instance document against the waterBody.xsd schema to verify the above requirement. The process may be using an appropriate software tool for validation or be a manual process that checks all definitions from the XML schema specification.

Abstract Test 43	
Test ID:	/conf/waterbody/boundaries
Requirement:	/req/waterbody/boundaries
Test purpose:	For each WaterBody space element, verify that if the space element is bounded by thematic surface boundaries using the property <i>core:boundary</i> (type: <i>core:AbstractSpaceBoundaryPropertyType</i>), each property contains exactly one surface element from Table 31 that is supported for the specific space element. If no surface element is supported, the space element is not bounded by thematic surface boundaries.

Test method: Inspect the instance document to verify the above requirement.

Annex B: Examples (Informative)

Various CityGML 3.0 test data sets are available in the [folder examples on the CityGML 3.0 GML Encoding GitHub](#) for each CityGML module, including those test data sets presented in this chapter.

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 the 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:

```
https://thingspeak.com/channels/64242/fields/1.json?start=2019-10-08T00:00:00Z&end=2019-10-09T00:00:00Z
```

[Listing 5](#) shows, how the Dynamizer makes use of this query to link the generic attribute *temperature* to the temperature observation of the ThingSpeak datastream with field ID 1.

Listing 5. Dynamizer defining a sensor connection to the ThingSpeak platform.

```
<bldg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer1">
      <dyn:attributeRef>
        <!--Single line without linebreaks and spaces-->
        //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:baseURL>https://thingspeak.com</dyn:baseURL>
          <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>
  <!-- The roof surface and building room from Listing 4 come here -->
</bldg:Building>
```

The Dynamizer specifies the following parameters:

- *attributeRef* defines an XPath expression which refers to the generic attribute *temperature* of the BuildingRoom feature with the ID *room1* that is to be overridden by sensor data from the specific ThingSpeak channel.
- *startTime* and *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>.
- *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>.
- *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 FROST Server is an open-source implementation of the OGC SensorThings API standard providing standardized interfaces to retrieve sensor data and observations. 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*.

Listing 6. Dynamizer defining a sensor connection to the FROST Server.

```
<bldg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer2">
      <dyn:attributeRef>
        <!--Single line without linebreaks and spaces-->
        //bldg:BuildingRoom[@gml:id='room1']/core:genericAttribute/
        gen:DoubleAttribute[gen:name='temperature']/gen:value
      </dyn:attributeRef>
      <dyn:startTime>2019-04-12T10:00:00.000Z</dyn:startTime>
      <dyn:endTime>2019-04-12T10:30:00.000Z</dyn:endTime>
      <dyn:sensorConnection>
        <dyn:SensorConnection>
          <dyn:connectionType>ogc_sta_1.0</dyn:connectionType>
          <dyn:observationProperty>Temperature</dyn:observationProperty>
          <dyn:uom>Celsius</dyn:uom>
          <dyn:datastreamID>1</dyn:datastreamID>
          <dyn:baseURL>http://127.0.0.1:8080/FROST-Server/v1.0</dyn:baseURL>
          <dyn:authType>none</dyn:authType>
          <dyn:linkToObservation>
            <!--Single line without linebreaks and spaces-->
            %baseUrl%/Datastreams(%datastreamID%)/
            Observations?$filter=during(phenomenonTime,%startTime%/%endTime%)
          </dyn:linkToObservation>
          <dyn:linkToSensorDescription>
            %baseUrl%/Datastreams(%datastreamID%)/Sensor
          </dyn:linkToSensorDescription>
          <dyn:sensorLocation xlink:href="#room1"/>
        </dyn:SensorConnection>
      </dyn:sensorConnection>
    </dyn:Dynamizer>
  </core:dynamizer>
  <!-- The roof surface and building room from Listing 4 come here -->
</bldg:Building>
```

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\(phenomeno nTime,2019-04-12T10:00:00.000Z/2019-04-12T10:30:00.000Z\).](http://127.0.0.1:8080/FROSTServer/v1.0/Datastreams(1)/Observations?$filter=during(phenomeno nTime,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](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.

```
<bldg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer3">
      <dyn:attributeRef>
        <!--Single line without linebreaks and spaces-->
        //bldg: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>
  <!-- The roof surface and building room from Listing 4 come here -->
</bldg: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:

http://api.wunderground.com/api/*****/conditions/q/pws:MyPWS1.json

[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.

```
<bldg:Building gml:id="building1">
  <core:dynamizer>
    <dyn:Dynamizer gml:id="dynamizer4">
      <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>2019-01-01T10:00:00.000Z</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>
    </dyn:Dynamizer>
  </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':

```
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.

```
<bldg: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>
  <!-- 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 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.

```
Surface_ID,Time,Uom,Diffuse,Direct,Global  
wall1,2015-01,kWh,1454.653,3315.214,4293.446  
wall1,2015-02,kWh,1866.883,4002.232,5563.502  
.....  
.....  
wall1,2015-12,kWh,1341.543,3001.412,4010.239  
wall2,2015-01,kWh,1313.344,3112.122,4109.742  
.....  
.....
```

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*.

Listing 12. Dynamizer using a TabulatedFileTimeseries object for referring to timeseries stored in an external CSV file using the column ids.

```
<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.

```
<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.

```
<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.

```
<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.

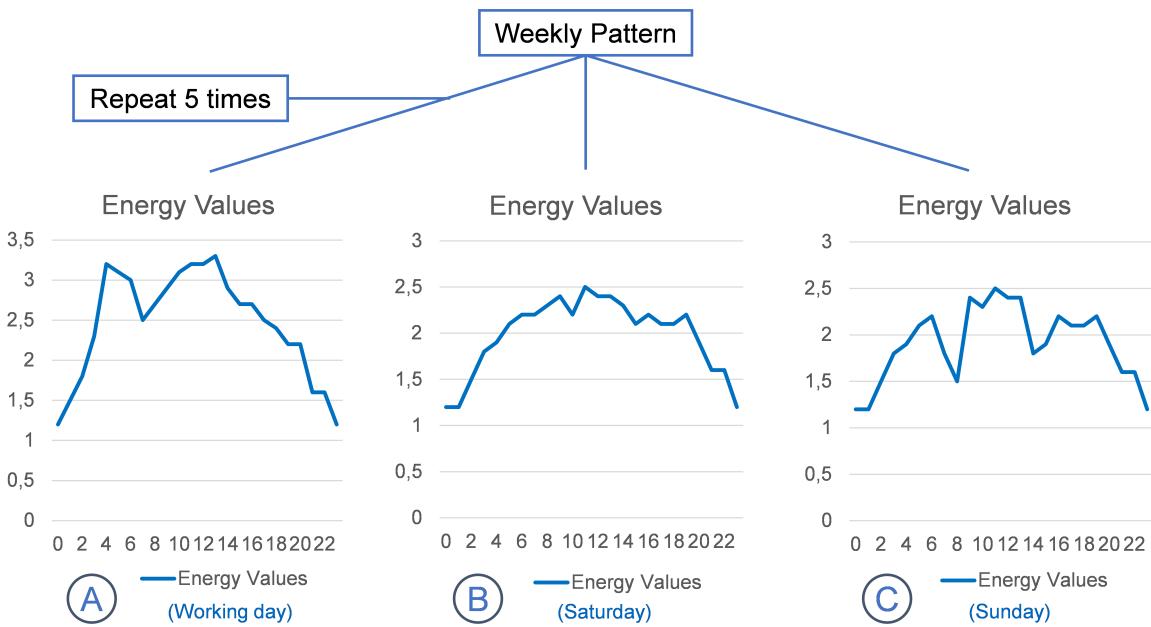


Figure 14. Example of a energy consumption pattern composed of atomic timeseries (source: [Chaturvedi 2021](#)).

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 'AAAAABC'). This is illustrated in Listing 16.

Listing 16. Dynamizer defining a CompositeTimeseries to represent a weekly pattern of energy consumption values.

```
<!-- Only the Dynamizer feature is shown in this listing -->
<dyn:Dynamizer gml:id="electricity_consumption">
  <dyn:attributeRef>
    <!-- XPath expression to the city object property to be overridden-->
  </dyn:attributeRef>
  <!-- ISO 8601 Week Date Representation showing absolute timestamps-->
  <dyn:startTime>2015-W01-01</dyn:startTime>
  <dyn:endTime>2015-W02-01</dyn:endTime>
  <dyn:dynamicData>
    <dyn:CompositeTimeseries>
      <!-- Component for weekdays-->
      <dyn:component>
        <dyn:TimeseriesComponent>
          <dyn:repetitions>5</dyn:repetitions>
          <dyn:timeseries>
            <dyn:GenericTimeseries gml:id="Weekdays">
              <!-- ISO 8601 Time Representation showing relative timestamps-->
              <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:timeValuePair>
```

```

<dyn:TimeValuePair>
    <dyn:timestamp>T00:00:00</dyn:timestamp>
    <dyn:doubleValue>1.32</dyn:doubleValue>
</dyn:TimeValuePair>
</dyn:timeValuePair>
<dyn:TimeValuePair>
    <dyn:timestamp>T01:00:00</dyn:timestamp>
    <dyn:doubleValue>1.41</dyn:doubleValue>
</dyn:TimeValuePair>
</dyn:timeValuePair>
<dyn:TimeValuePair>
    <dyn:timestamp>T02:00:00</dyn:timestamp>
    <dyn:doubleValue>1.53</dyn:doubleValue>
</dyn:TimeValuePair>
</dyn:timeValuePair>
<!-- ... values for all 24 hours in a weekday ... -->
</dyn:GenericTimeseries>
</dyn:timeseries>
</dyn:TimeseriesComponent>
</dyn:component>
<!-- Component for Saturday-->
<dyn:component>
    <dyn:TimeseriesComponent>
        <dyn:repetitions>1</dyn:repetitions>
        <dyn:timeseries>
            <dyn:GenericTimeseries gml:id="Saturday">
                <!-- ISO 8601 Time Representation showing relative timestamps-->
                <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:TimeValuePair>
                    <dyn:TimeValuePair>
                        <dyn:timestamp>T00:00:00</dyn:timestamp>
                        <dyn:doubleValue>1.39</dyn:doubleValue>
                    </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:timestamp>T02:00:00</dyn:timestamp>
                <dyn:doubleValue>1.52</dyn:doubleValue>
            </dyn:TimeValuePair>
        </dyn:timeseries>
    </dyn:TimeseriesComponent>
</dyn:component>

```

```

</dyn:timeValuePair>
    <!-- ... values for all 24 hours in a weekday ... -->
</dyn:GenericTimeseries>
</dyn:timeseries>
</dyn:TimeseriesComponent>
</dyn:component>
<!-- Component for Sunday-->
<dyn:component>
    <dyn:TimeseriesComponent>
        <dyn:repetitions>1</dyn:repetitions>
        <dyn:timeseries>
            <dyn:GenericTimeseries gml:id="Sunday">
                <!-- ISO 8601 Time Representation showing relative timestamps-->
                <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:timeValuePair>
                    <dyn:TimeValuePair>
                        <dyn:timestamp>T00:00:00</dyn:timestamp>
                        <dyn:doubleValue>1.30</dyn:doubleValue>
                    </dyn:TimeValuePair>
                </dyn:timeValuePair>
                <dyn:timeValuePair>
                    <dyn:TimeValuePair>
                        <dyn:timestamp>T01:00:00</dyn:timestamp>
                        <dyn:doubleValue>1.46</dyn:doubleValue>
                    </dyn:TimeValuePair>
                </dyn:timeValuePair>
                <dyn:timeValuePair>
                    <dyn:TimeValuePair>
                        <dyn:timestamp>T02:00:00</dyn:timestamp>
                        <dyn:doubleValue>1.59</dyn:doubleValue>
                    </dyn:TimeValuePair>
                </dyn:timeValuePair>
                <!-- ... values for all 24 hours in a weekday ... -->
            </dyn:GenericTimeseries>
        </dyn:timeseries>
    </dyn:TimeseriesComponent>
</dyn:component>
</dyn:CompositeTimeseries>
</dyn:dynamicData>
</dyn:Dynamizer>

```

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.

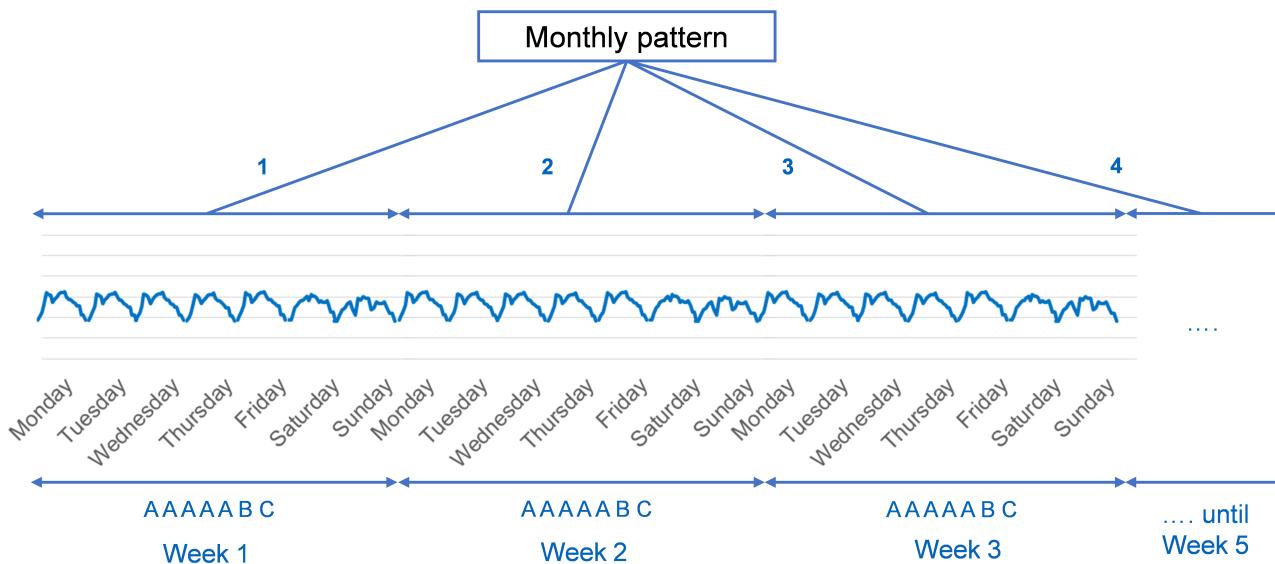


Figure 15. Example of a monthly pattern composed of individual CompositeTimeseries defined by complex energy consumption pattern comosed of atomic timeseries (source: [Chaturvedi 2021](#)).

[Listing 16](#) 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 described in [Beil et al. 2021](#) and are illustrated in the following.

The CityGML and LAZ data sets presented here are available on the [CityGML 3.0 GML Encoding GitHub](#).

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 17](#) 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.

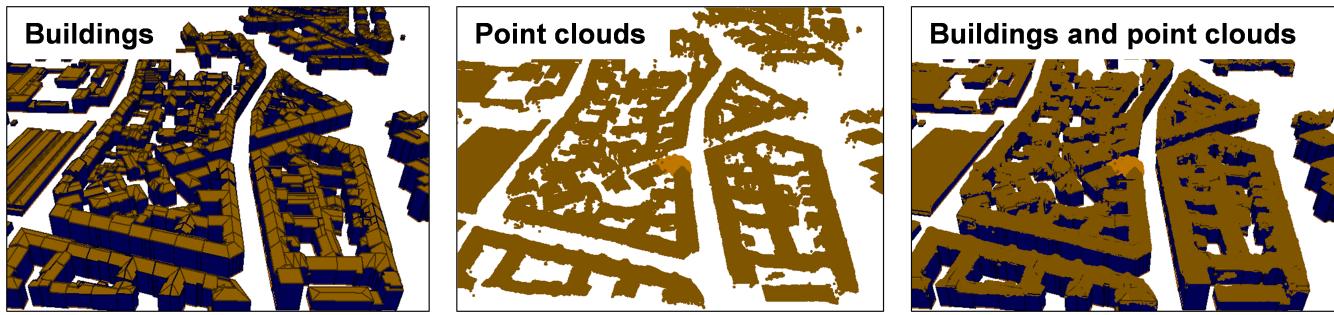


Figure 16. Visualization of the inline representation of points clouds.

Listing 17. The point cloud of a building is embedded in the Building feature.

```
<?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 18](#), where the building feature references the LAZ file *DEBY_LOD2_4903217.laz* that contains the points for this building (illustrated in [Figure 17](#)).

Listing 18. 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-8ddc-4780566814a5">
          <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 and

all points with the corresponding value in the attribute *point_source_id*. In Listing 19, 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 19. 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_5e7144be-258f-4d58-8f0c-
bd10ccb1387a5">

<pcl:pointFile>..\pointcloud\4467_5331_40_bDOM_classified.laz?idattr=point_source_id&a-
mp;id=132</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>
```

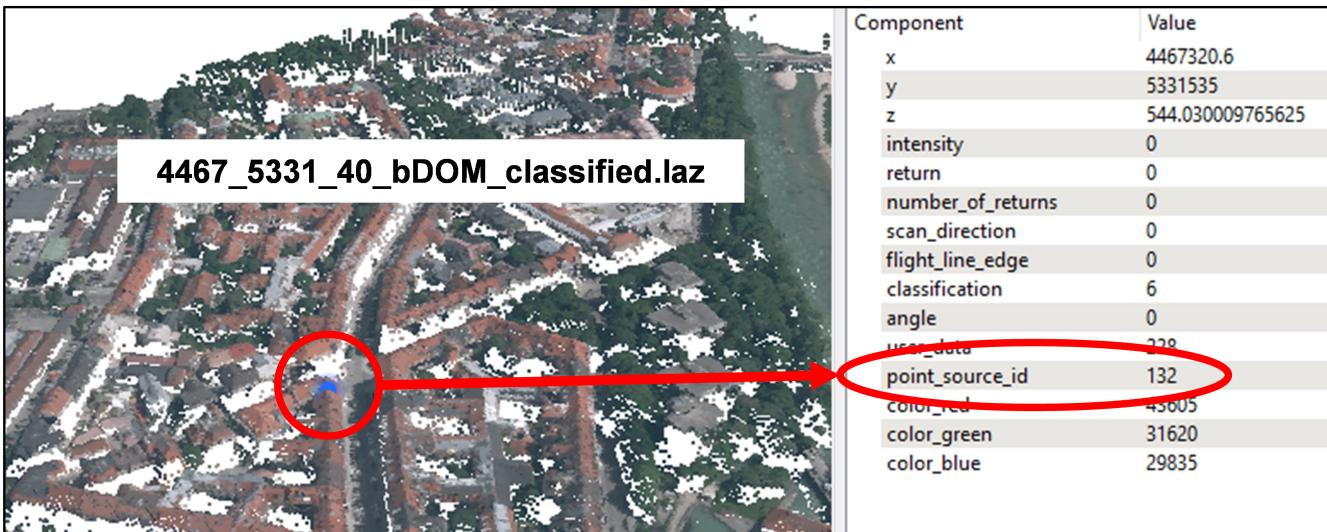


Figure 18. Visualization of an external LAZ file containing amongst others the referenced points of a specific building.

B.3. Versioning Examples

Different versions of city models can be managed with CityGML 3.0 in two ways, 1) using XLinks and identifiers of features, and 2) using the concepts *Version* and *VersionTransition* defined in the Versioning module. Both methods are presented in the following.

B.3.1. 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 is created which receives a new *gml:id*.
2. All subordinate features, i.e. the features that are part of the top-level feature, remain unchanged and are 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 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 gml:id. Since the subordinate feature BuildingPart remains unchanged, the new Building version references the unchanged BuildingPart feature.

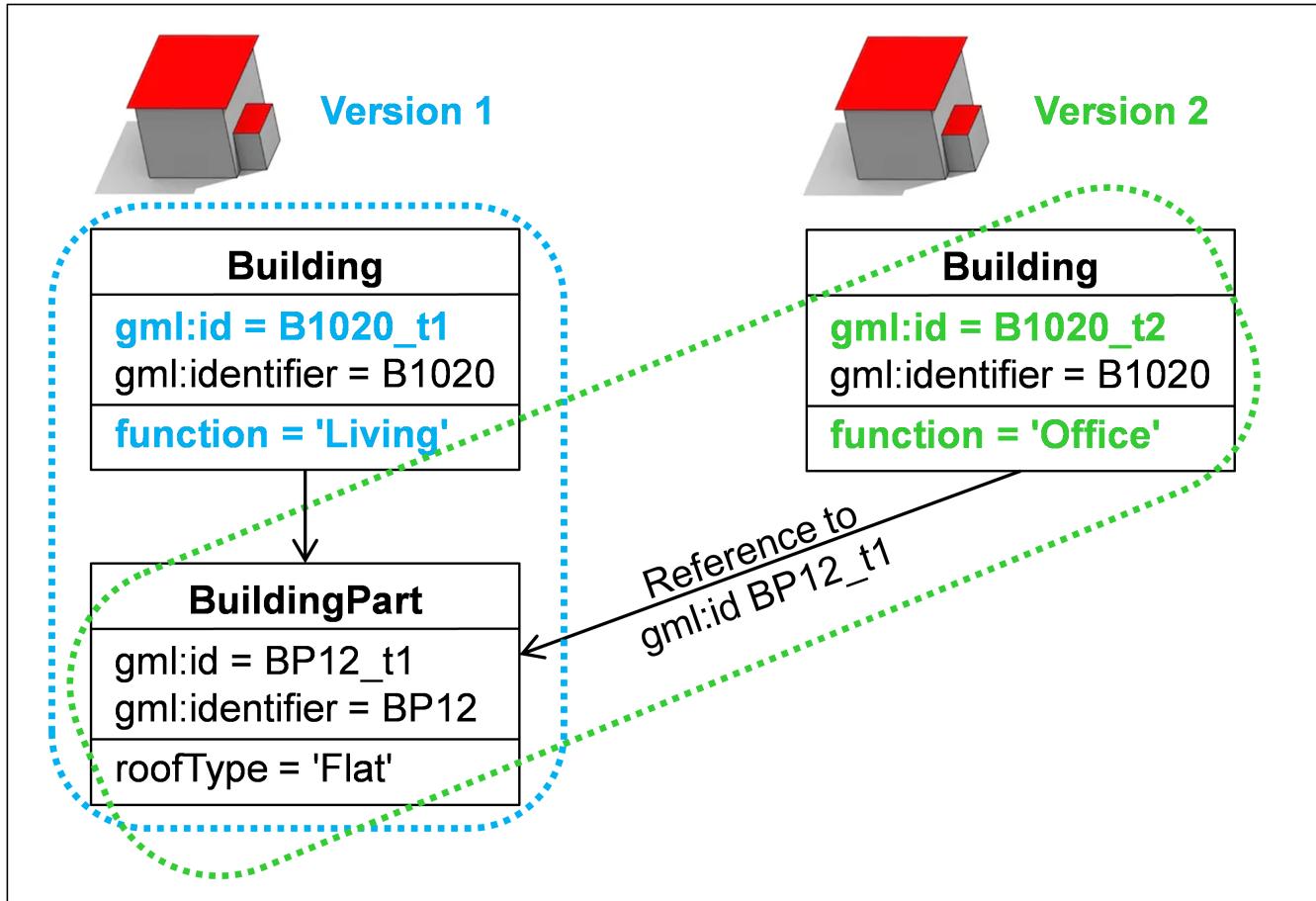


Figure 19. Version management when only the top-level feature changes.

Rule 2: Only a subordinate feature changes

1. When only a subordinate feature changes, a new version of this subordinate feature is 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 is adapted to this new gml:id as well, which means that also a new version of the superordinate feature is created.
3. Changes to these XLink references and creation of new superordinate features are applied transitively to all superordinate features up to the top-level feature.
4. All other subordinate features that do not change are 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.

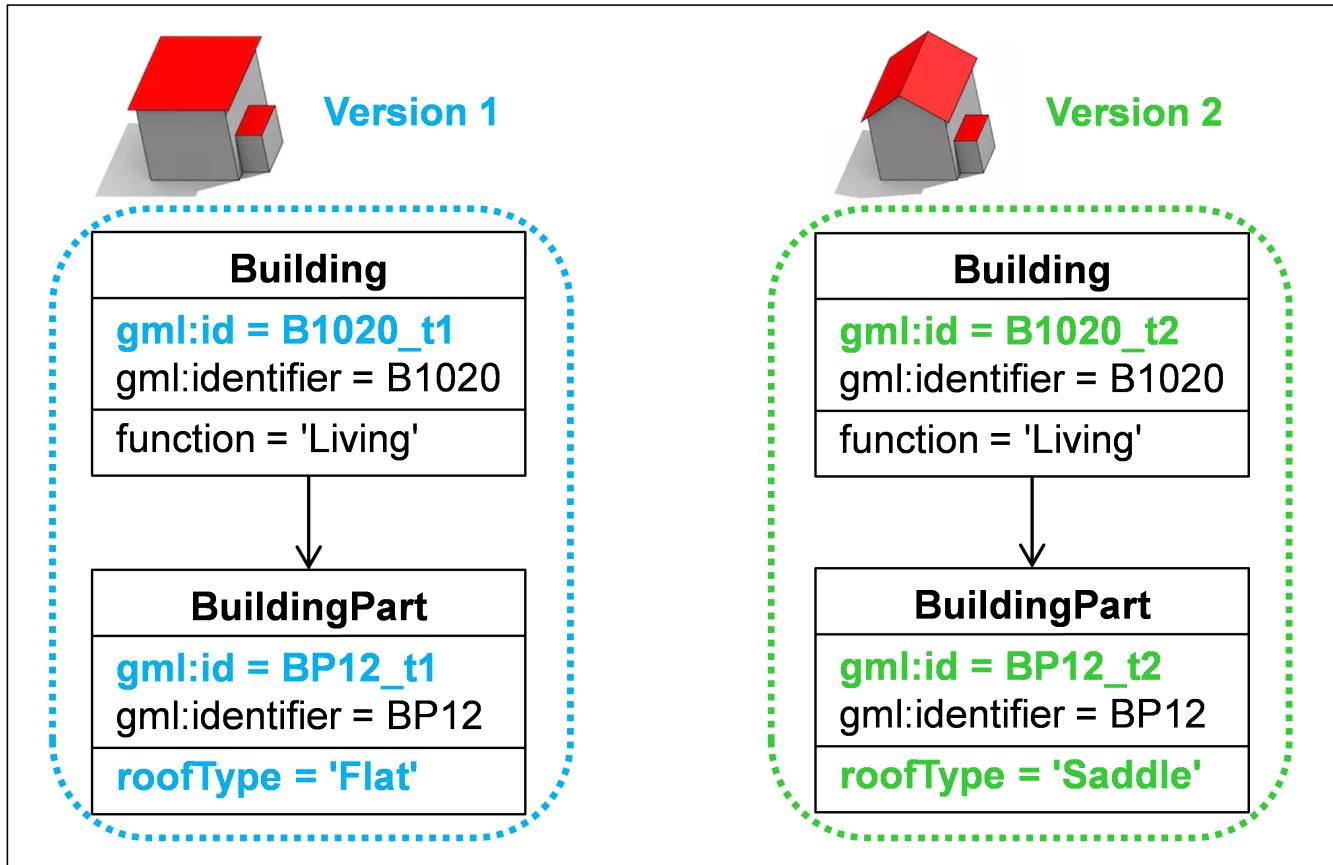


Figure 20. Version management when only a subordinate feature changes that is directly referenced from the top-level feature.

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.

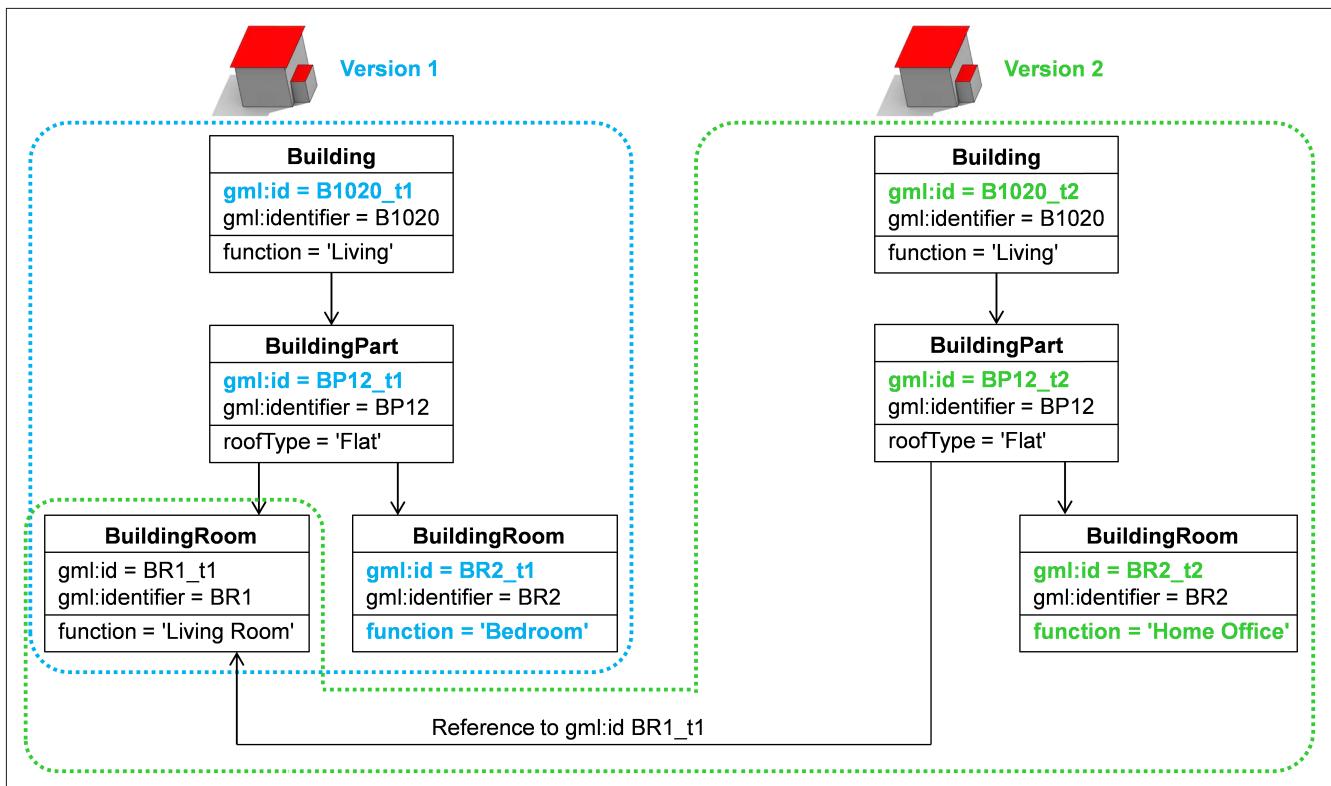


Figure 21. Version management when only a subordinate feature changes that is part of an aggregation hierarchy.

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 are applied jointly.

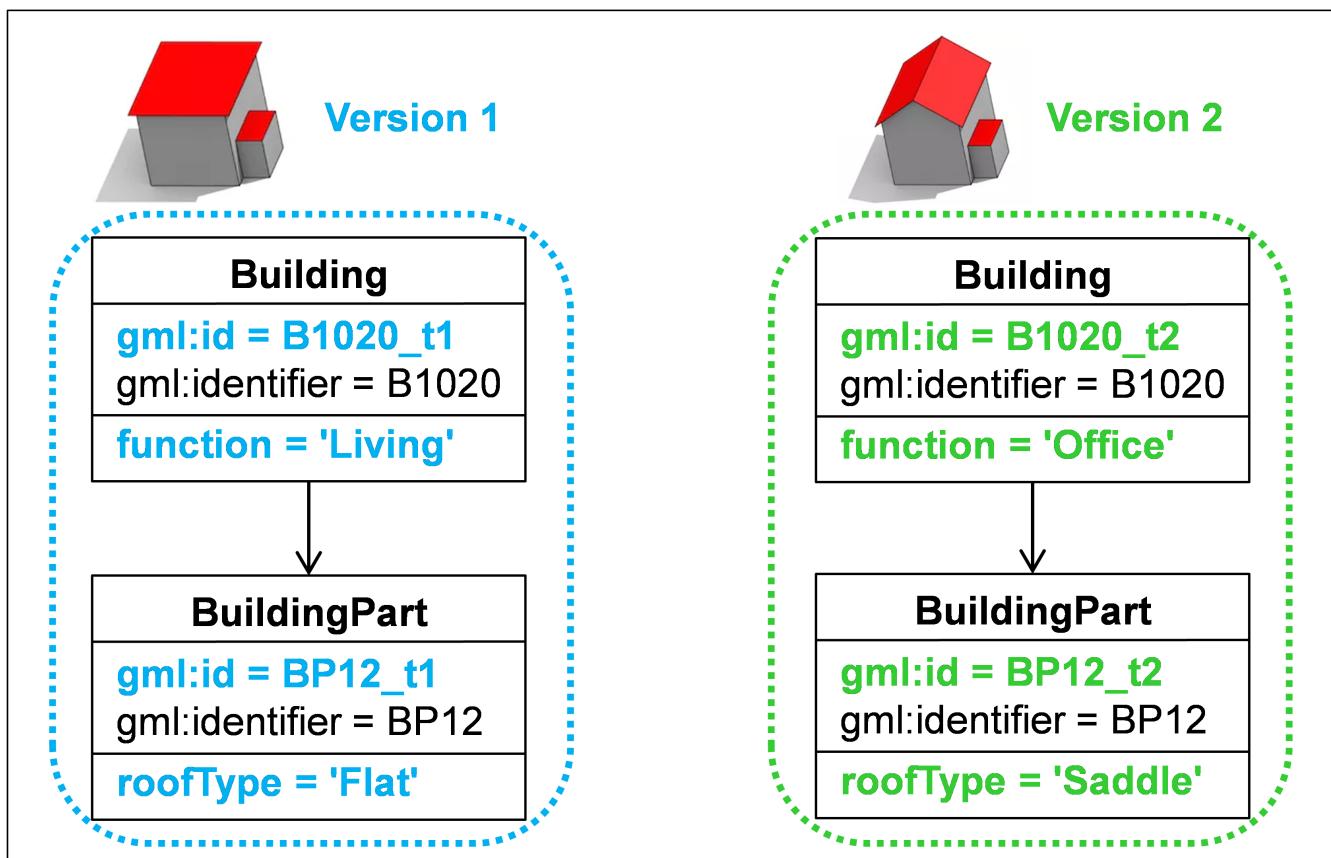


Figure 22. Use of Rule 3 in case a top-level feature and a corresponding subordinate feature changes.

The use of rule 3 is illustrated in [Figure 22](#). 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 20](#) represents this example in GML.

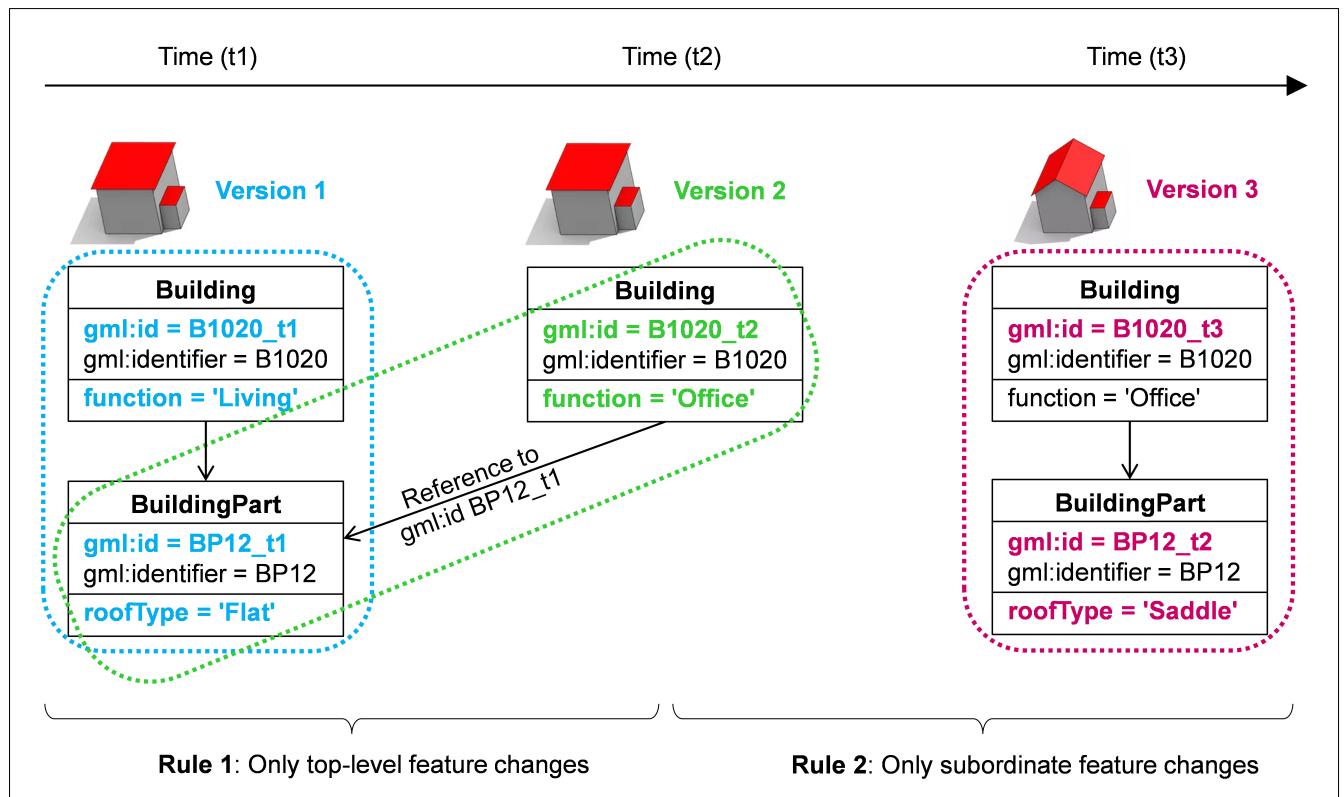


Figure 23. Successive version management over a period of time.

Listing 20. GML instance document for the example on successive version management over a period of time illustrated in Figure 23.

```

<?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>

```

B.3.2. Version management using Versions and VersionTransitions

The example in [Figure 24](#) illustrates four Building features (b1, b2-1, b2-2, and b3) represented across two Versions (v1 and v2). A VersionTransition (vt1) of the type *historicalSuccession* stores the changes between each Building through three 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 21 represents this example in GML.

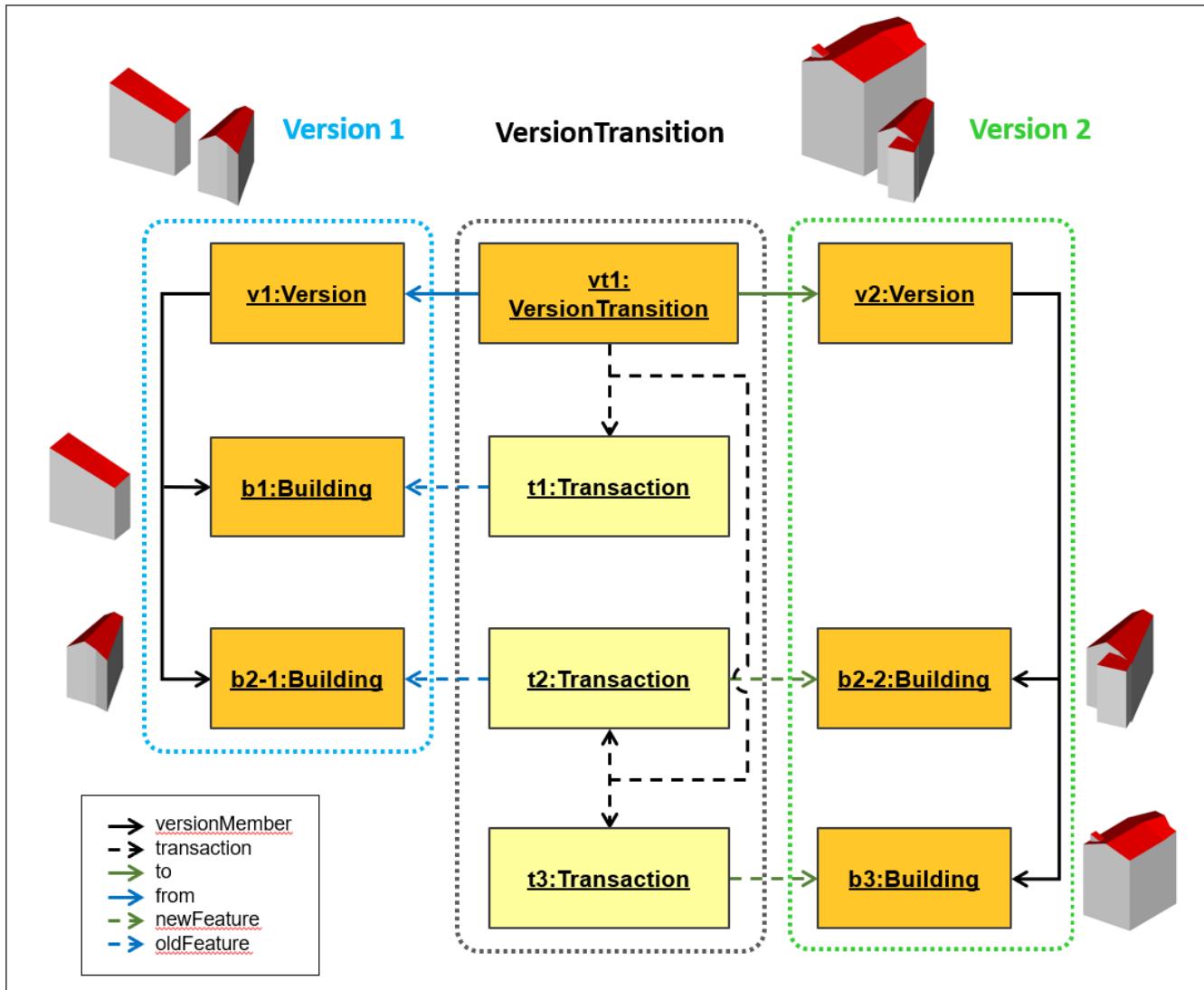


Figure 24. Successive version management using Versions and VersionTransitions.

Listing 21. 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:versionMember>
<core:versionTransitionMember>
    <vers:VersionTransition>
        <vers:reason>To exemplify a version transition between two city
snapshots</vers:reason>
        <vers:type>historicalSuccession</vers:type>
        <vers:from xlink:href="#v1"/>
        <vers:to xlink:href="#v2"/>
        <vers:transaction>
            <vers:Transaction>
                <vers:type>delete</vers:type>
                <vers:oldFeature xlink:href="#b1"/>
            </vers:Transaction>
        </vers:transaction>
        <vers:transaction>
            <vers:Transaction>
                <vers:type>replace</vers:type>
                <vers:oldFeature xlink:href="#b2-1"/>
                <vers:newFeature xlink:href="#b2-2"/>
            </vers:Transaction>
        </vers:transaction>
        <vers:transaction>
            <vers:Transaction>
                <vers:type>insert</vers:type>
                <vers:newFeature xlink:href="#b3"/>
            </vers:Transaction>
        </vers:transaction>
    </vers:VersionTransition>
</core:versionTransitionMember>
<core:cityObjectMember>
    <bldg:Building gml:id="b1">
        <gml:identifier codeSpace="http://example.org">b1</gml:identifier>
    </bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
    <bldg:Building gml:id="b2-1">
        <gml:identifier codeSpace="http://example.org">b2</gml:identifier>
    </bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
    <bldg:Building gml:id="b2-2">
        <gml:identifier codeSpace="http://example.org">b2</gml:identifier>
    </bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
    <bldg:Building gml:id="b3">
        <gml:identifier codeSpace="http://example.org">b3</gml:identifier>
    </bldg:Building>
</core:cityObjectMember>
</core:CityModel>

```

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 on the [CityGML 3.0 GML Encoding GitHub](#), additional examples are also available on the [OASIS documentation repository](#).

Example 1: Structured address and multipoint

xAL provides a rich vocabulary for structuring addresses in GML. Listing 22 shows how an address in Germany can be modeled using xAL and a multipoint:

Listing 22. A GML example of a structured German xAL building address with a GML MultiPoint.

```
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
  <core:cityObjectMember>
    <bldg:Building gml:id="DEBY_LOD2_5744682">
      <bldg:address>
        <core:Address>
          <!--
            Am Hang 15      street number and name
            85665 Moosach  postcode and locality
            Germany         country
          -->
        <core:xalAddress>
          <xAL:Address>
            <xAL:Country>
              <xAL:NameElement xAL:NameType="Name">Germany</xAL:NameElement>
            </xAL:Country>
            <xAL:Locality xAL:Type="Town">
              <xAL:NameElement xAL:NameType="Name">Moosach</xAL:NameElement>
            </xAL:Locality>
            <xAL:Thoroughfare xAL:Type="Street">
              <xAL:NameElement xAL:NameType="NameAndType">Am Hang</xAL:NameElement>
              <xAL:Number xAL:Type="Number">15</xAL:Number>
            </xAL:Thoroughfare>
            <xAL:PostCode>
              <xAL:Identifier>85665</xAL:Identifier>
            </xAL:PostCode>
          </xAL:Address>
        </core:xalAddress>
        <!-- WGS84 48.03616458630443, 11.873209628265858 -->
        <core:multiPoint>
          <gml:MultiPoint srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:pointMember>
              <gml:Point>
                <gml:pos>48.03616458630443 11.873209628265858</gml:pos>
              </gml:Point>
            </gml:pointMember>
          </gml:MultiPoint>
        </core:multiPoint>
      </core:Address>
    </bldg:address>
  </bldg:Building>
</core:cityObjectMember>
</core:CityModel>
```

Example 2: Free text address

More freeform addresses can also be declared using FreeTextAddresses as shown in [Listing 23](#):

Listing 23. A GML example of a free text xAL building address based on the National Opera of Lyon, France.

```
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel ... Namespaces and Schema location ... >
  <core:cityObjectMember>
    <bldg:Building gml:id="OPERA LYON_45">
      <bldg:address>
        <core:Address>
          <!--
            Opéra National de Lyon      additional geographical information
            1 Place de la Comédie       street number and name
            69001 Lyon                  postcode and locality
            France                      country
          -->
        <core:xalAddress>
          <xAL:Address>
            <xAL:FreeTextAddress>
              <xAL:AddressLine>Opéra National de Lyon</xAL:AddressLine>
              <xAL:AddressLine>1 Place de la Comédie</xAL:AddressLine>
              <xAL:AddressLine>69001 Lyon</xAL:AddressLine>
              <xAL:AddressLine>France</xAL:AddressLine>
            </xAL:FreeTextAddress>
          </xAL:Address>
        </core:xalAddress>
      </core:Address>
    </bldg:address>
  </bldg:Building>
</core:cityObjectMember>
</core:CityModel>
```

B.5. Application Domain Extension (ADE) Examples

The CityGML 3.0 Conceptual Model continues to offer 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 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 from 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 (available [here](#)). 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. Both methods, the 'hook' mechanism for augmenting predefined CityGML feature classes with additional ADE properties and subclassing for adding new application-specific feature types to CityGML, 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. *BuildingUndergroundClassValue*, *BuildingUndergroundFunctionValue*, and *BuildingUndergroundUsageValue* show how to define code lists for ADE class attributes.
2. *EnergyPerformanceCertification* is an example of how to define an ADE data type, which can comprise a set of simple attributes.
3. *BuildingProperties* is a subclass of the 'hook' data type *ADEOfAbstractBuilding* for augmenting the predefined CityGML class *AbstractBuilding* with additional properties.
4. *Facility* and its subclasses *ElectricalAppliance* and *LightingFacility* are object types, which are not derived from any predefined CityGML class.
5. *IndustrialBuilding* and *MovingConstruction* are two top-level feature type classes derived from the predefined CityGML top-level classes *Building* and *OtherConstruction*, respectively.
6. *IndustrialBuildingPart* and *SolarRoofSurface* are two examples of non-top-level feature type classes, which are derived from the CityGML feature type classes *BuildingPart* and *RoofSurface*.
7. *AbstractBuildingUnderground* and its subclasses *BuildingUnderground* and *BuildingUndergroundPart* are derived from the space class *AbstractOccupiedSpace* in the CityGML Core Module, and comprise a couple of new thematic and spatial properties in addition to the properties defined in the superclass. In addition, the class *AbstractBuildingUnderground* defines a new geometry which is not available in the CityGML Core module.

A corresponding XML schema derived from the UML Conceptual Model is shown in [Listing 24](#). An example ADE dataset is represented in [Listing 25](#).

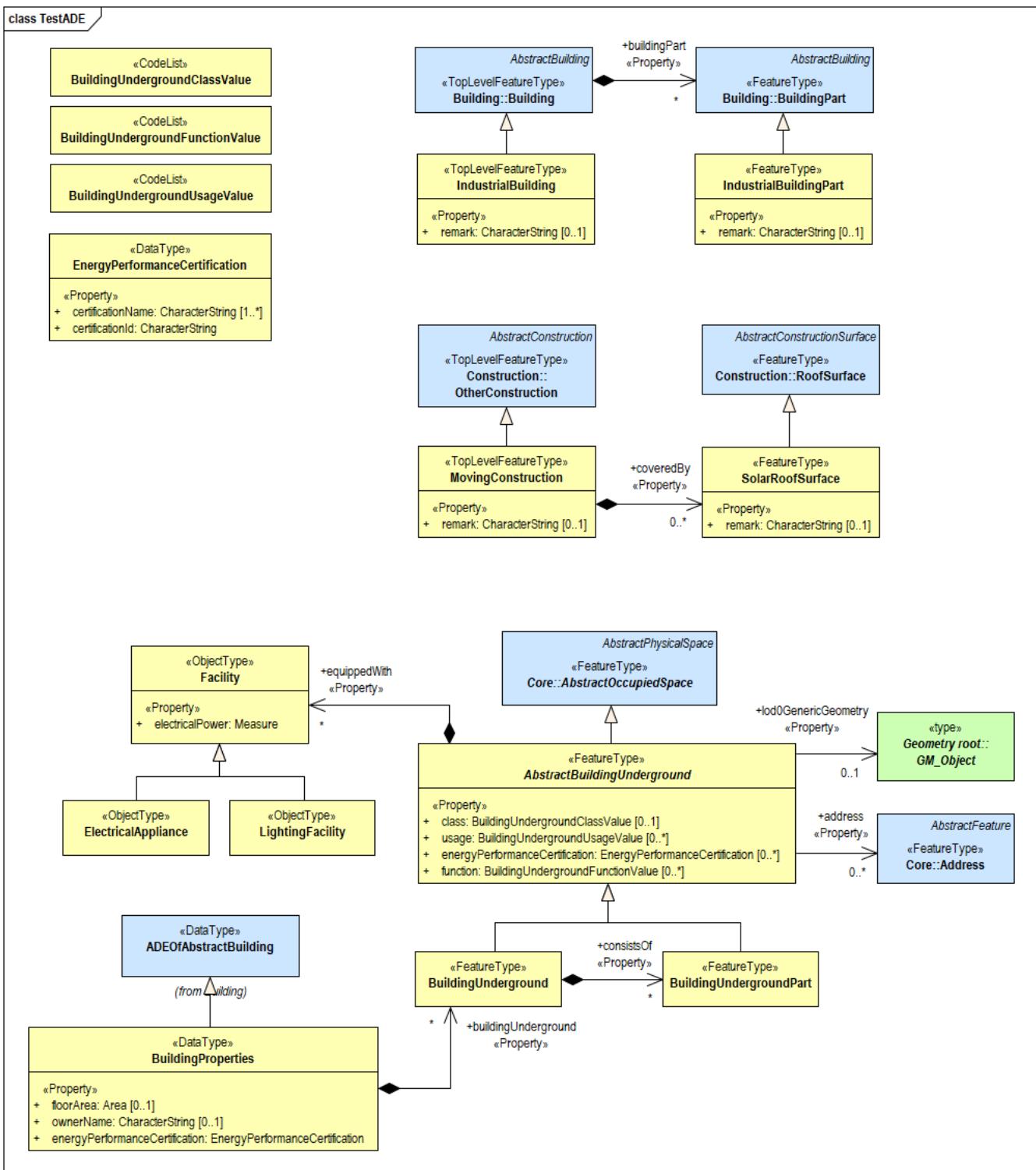


Figure 25. UML model of the Test ADE

Listing 24. XML schema of the Test ADE model in Figure 25

```

<?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"/>
```

```

<import namespace="http://www.opengis.net/citygml/building/3.0"
schemaLocation="../schema/building.xsd"/>
<import namespace="http://www.opengis.net/citygml/construction/3.0"
schemaLocation="../schema/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/--&gt;
&lt;element abstract="true" name="AbstractBuildingUnderground"
substitutionGroup="core:AbstractOccupiedSpace"
type="test:AbstractBuildingUndergroundType"/&gt;
&lt;complexType abstract="true" name="AbstractBuildingUndergroundType"&gt;
&lt;complexContent&gt;
&lt;extension base="core:AbstractOccupiedSpaceType"&gt;
&lt;sequence&gt;
&lt;element minOccurs="0" name="class" type="gml:CodeType"/&gt;
&lt;element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType"/&gt;
&lt;element maxOccurs="unbounded" minOccurs="0"
name="energyPerformanceCertification"
type="test:EnergyPerformanceCertificationPropertyType"/&gt;
&lt;element minOccurs="0" name="lod0GenericGeometry"
type="gml:GeometryPropertyType"/&gt;
&lt;element maxOccurs="unbounded" minOccurs="0" name="address"
type="core:AddressPropertyType"/&gt;
&lt;element maxOccurs="unbounded" minOccurs="0" name="equippedWith"&gt;
&lt;complexType&gt;
&lt;complexContent&gt;
&lt;extension base="gml:AbstractMemberType"&gt;
&lt;sequence minOccurs="0"&gt;
&lt;element ref="test:Facility"/&gt;
&lt;/sequence&gt;
&lt;attributeGroup ref="gml:AssociationAttributeGroup"/&gt;
&lt;/extension&gt;
&lt;/complexContent&gt;
&lt;/complexType&gt;
&lt;/element&gt;
&lt;element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType"/&gt;
&lt;/sequence&gt;
&lt;/extension&gt;
&lt;/complexContent&gt;
&lt;/complexType&gt;
&lt;complexType name="AbstractBuildingUndergroundPropertyType"&gt;
&lt;sequence minOccurs="0"&gt;
&lt;element ref="test:AbstractBuildingUnderground"/&gt;
&lt;/sequence&gt;
&lt;attributeGroup ref="gml:AssociationAttributeGroup"/&gt;
&lt;attributeGroup ref="gml:OwnershipAttributeGroup"/&gt;
&lt;/complexType&gt;
&lt;element name="BuildingProperties" substitutionGroup="bldg:ADEofAbstractBuilding"
type="test:BuildingPropertiesType"/&gt;
</pre>

```

```

<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="BuildingProperties.PropertyType">
  <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>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

```

</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">

```

```

<sequence>
    <element ref="test:EnergyPerformanceCertification"/>
</sequence>
</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"/>

```

```
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="SolarRoofSurface" substitutionGroup="con:RoofSurface"
type="test:SolarRoofSurfaceType"/>
<complexType name="SolarRoofSurfaceType">
<complexContent>
<extension base="con:RoofSurfaceType">
<sequence>
<element minOccurs="0" name="remark" type="string"/>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="SolarRoofSurfacePropertyType">
<sequence minOccurs="0">
<element ref="test:SolarRoofSurface"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>
```

Listing 25. Example dataset implementing the illustrated Test ADE schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CityModel xmlns="http://www.opengis.net/citygml/3.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:bldg="http://www.opengis.net/citygml/building/3.0"
  xmlns:test="http://www.citygml.org/ade/TestADE/2.0">
  <cityObjectMember>
    <test:IndustrialBuilding>
      <bldg:class>1100</bldg:class>
      <bldg:roofType>1030</bldg:roofType>
      <bldg:storeysAboveGround>5</bldg:storeysAboveGround>
      <bldg:storeysBelowGround>0</bldg:storeysBelowGround>
      <bldg:storeyHeightsAboveGround uom="#m">2.0</bldg:storeyHeightsAboveGround>
      <bldg:storeyHeightsBelowGround uom="#m">0.0</bldg:storeyHeightsBelowGround>
      <bldg:adeOfAbstractBuilding>
        <test:BuildingProperties>
          <test:ownerName>Smith</test:ownerName>
          <test:energyPerformanceCertification>
            <test:EnergyPerformanceCertification>
              <test:certificationName>certName</test:certificationName>
              <test:certificationId>certId</test:certificationId>
            </test:EnergyPerformanceCertification>
          </test:energyPerformanceCertification>
          <test:buildingUnderground>
            <test:BuildingUnderground>
              <test:lod0GenericGeometry>
                <gml:Polygon>
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList srsDimension="3">6.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 8.0
0.0 6.0 8.0 0.0 6.0 0.0 0.0</gml:posList>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:Polygon>
                </test:lod0GenericGeometry>
                <test:equippedWith>
                  <test:LightingFacility>
                    <test:electricalPower uom="W">4000.0</test:electricalPower>
                  </test:LightingFacility>
                </test:equippedWith>
              </test:BuildingUnderground>
            </test:buildingUnderground>
          </test:BuildingProperties>
        </bldg:adeOfAbstractBuilding>
        <test:remark>remark</test:remark>
      </test:IndustrialBuilding>
    </cityObjectMember>
  </CityModel>
```

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 the actual UML model, 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 [Ishimaru et al. 2020](#) and [Akahoshi et al. 2020](#) for more information about the Urban Planning ADE.

Step 1: Designing the ADE as UML conceptual model

This example shows how to represent the Urban Object (namespace: uro) module from CityGML 2.0 compliant to CityGML 3.0 and how to inject additional properties to the predefined CityGML *Building* feature type. In order to allow using multiple distinct ADEs with 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, as was recommended in CityGML 2.0.

The hook has been implemented in 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 model for the Urban Object module has been implemented as described in Requirement 50 in [section 9.5 of the 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.

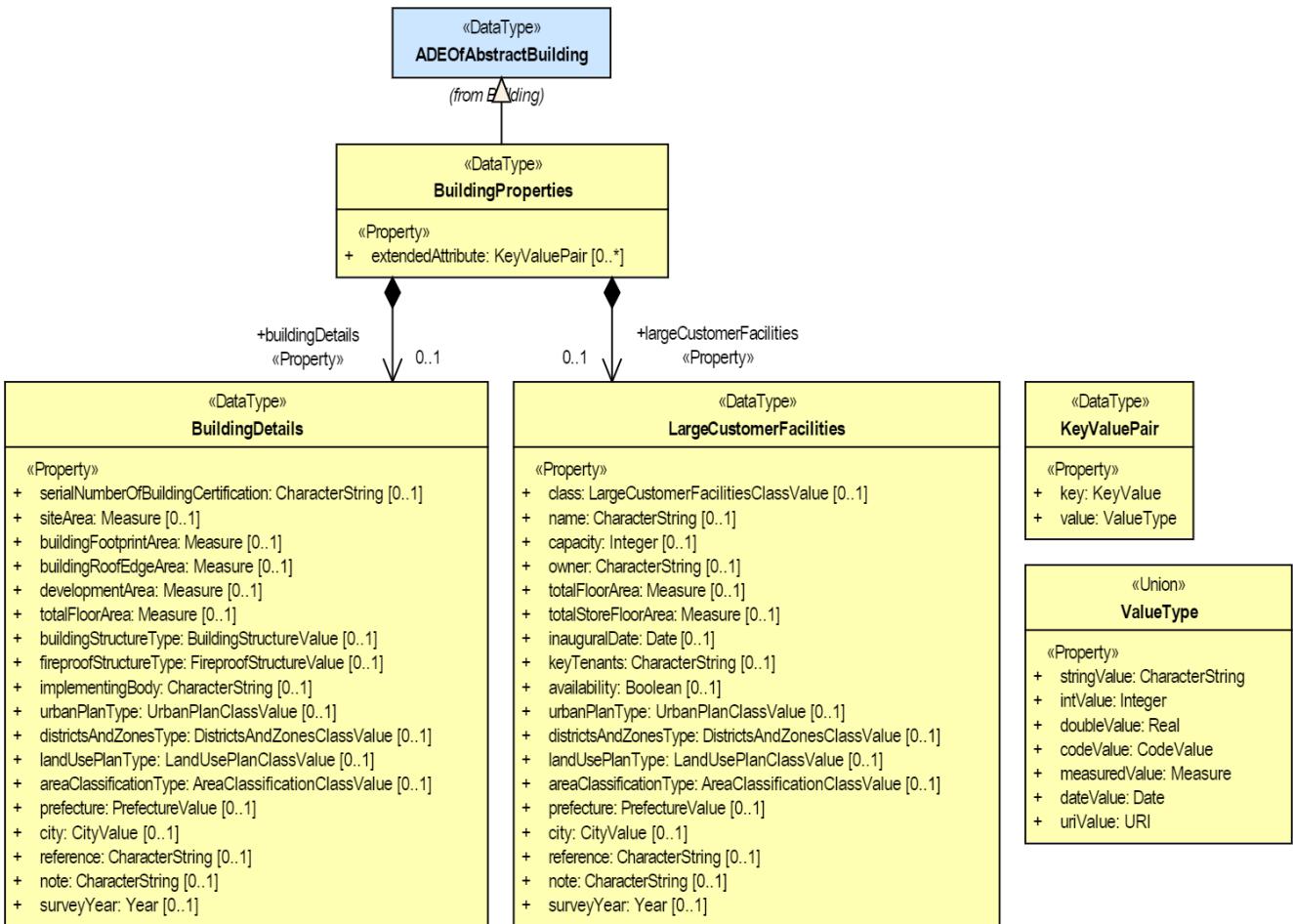


Figure 26. UML model of the Urban Object module from the Urban Planning ADE

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 26](#) and [the derived XML schema](#) from the Enterprise Architect UML model is illustrated in [Listing 27](#).

Listing 26. ShapeChange configuration file used for converting the ADE UML model to an XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<ShapeChangeConfiguration 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"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    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"/>
    <rules>
      <EncodingRule name="citygml" extends="iso19136_2007">
        <rule name="rule-xsd-rel-association-classes"/>
        <rule name="rule-xsd-cls-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-cls-union-asGroup"/>
      </EncodingRule>
    </rules>
    <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"/>
    </xmlNamespaces>
  </TargetXmlSchema>
</targets>

```

```

<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="http://docs.oasis-open.org/ciq/v3.0/cs02/xsd/default/xsd/xAL.xsd"/>
</xmlNamespaces>
<xi:include
href="http://shapechange.net/resources/config/StandardMapEntries.xml"/>
<xsdMapEntries>
<XsdMapEntry type="URI" xsdEncodingRules="iso19136_2007_gml33"
xmlPropertyType="anyURI" xmlType="anyURI" xmlTypeType="simple"
xmlTypeContent="simple"/>
</xsdMapEntries>
</TargetXmlSchema>
<Target
class="de.interactive_instruments.ShapeChange.Target.Codelists.CodelistDictionaries"
mode="disabled">
<targetParameter name="outputDirectory" value="..../codelists"/>
</Target>
</targets>
</ShapeChangeConfiguration>

```

Listing 27. XML schema document of the ADE derived from UML model

```

<?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="https://www.chisou.go.jp/tiiki/toshisaisei/itoshisaisei/iur/uro/1.5"
version="1.5">
    <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"/>
            <element minOccurs="0" name="note" type="string"/>
            <element minOccurs="0" name="surveyYear" type="gYear"/>
        </sequence>
    </complexType>
    <complexType name="BuildingDetails.PropertyType">
        <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:BuildingDetails.PropertyType"/>
                    <element minOccurs="0" name="largeCustomerFacilities"
type="uro:LargeCustomerFacilities.PropertyType"/>
                    <element maxOccurs="unbounded" minOccurs="0" name="extendedAttribute"
type="uro:KeyValuePair.PropertyType"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```

```

</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>
<element name="KeyValuePair" substitutionGroup="gml:AbstractObject"
type="uro:KeyValuePairType"/>
<complexType name="KeyValuePairType">
  <sequence>
    <element name="key" type="gml:CodeType"/>
    <group ref="uro:ValueTypeGroup"/>
  </sequence>
</complexType>
<complexType name="KeyValuePairPropertyType">
  <sequence>
    <element ref="uro:KeyValuePair"/>
  </sequence>
</complexType>
<element name="LandUseProperties" substitutionGroup="luse:ADEOfLandUse"
type="uro:LandUsePropertiesType"/>
<complexType name="LandUsePropertiesType">
  <complexContent>
    <extension base="luse:ADEOfLandUseType">
      <sequence>
        <element minOccurs="0" name="nominalArea" type="gml:MeasureType"/>
        <element minOccurs="0" name="ownerType" type="gml:CodeType"/>
        <element minOccurs="0" name="owner" type="string"/>
        <element minOccurs="0" name="areaInSquareMeter" type="gml:MeasureType"/>
        <element minOccurs="0" name="areaInHa" type="gml:MeasureType"/>
        <element minOccurs="0" name="urbanPlanType" type="gml:CodeType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<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"/>
<element minOccurs="0" name="note" type="string"/>
<element minOccurs="0" name="surveyYear" type="gYear"/>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="LandUsePropertiesPropertyType">
<sequence>
<element ref="uro:LandUseProperties"/>
</sequence>
</complexType>
<element name="LargeCustomerFacilities" substitutionGroup="gml:AbstractObject"
type="uro:LargeCustomerFacilitiesType"/>
<complexType name="LargeCustomerFacilitiesType">
<sequence>
<element minOccurs="0" name="class" type="gml:CodeType"/>
<element minOccurs="0" name="name" type="string"/>
<element minOccurs="0" name="capacity" type="integer"/>
<element minOccurs="0" name="owner" type="string"/>
<element minOccurs="0" name="totalFloorArea" type="gml:MeasureType"/>
<element minOccurs="0" name="totalStoreFloorArea" type="gml:MeasureType"/>
<element minOccurs="0" name="inauguralDate" type="date"/>
<element minOccurs="0" name="keyTenants" type="string"/>
<element minOccurs="0" name="availability" type="boolean"/>
<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"/>
<element minOccurs="0" name="note" type="string"/>
<element minOccurs="0" name="surveyYear" type="gYear"/>
</sequence>
</complexType>
<complexType name="LargeCustomerFacilitiesPropertyType">
<sequence>
<element ref="uro:LargeCustomerFacilities"/>
</sequence>
</complexType>
<element name="RoadProperties" substitutionGroup="tran:ADEOfRoad"
type="uro:RoadPropertiesType"/>
<complexType name="RoadPropertiesType">
<complexContent>
<extension base="tran:ADEOfRoadType">

```

```

<sequence>
    <element minOccurs="0" name="width" type="gml:LengthType"/>
    <element minOccurs="0" name="widthType" type="gml:CodeType"/>
    <element minOccurs="0" name="trafficVolume"
type="uro:TrafficVolumePropertyType"/>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="RoadPropertiesPropertyType">
    <sequence>
        <element ref="uro:RoadProperties"/>
    </sequence>
</complexType>
<element name="TrafficVolume" substitutionGroup="gml:AbstractObject"
type="uro:TrafficVolumeType"/>
<complexType name="TrafficVolumeType">
    <sequence>
        <element minOccurs="0" name="weekday12hourTrafficVolume" type="integer"/>
        <element minOccurs="0" name="weekday24hourTrafficVolume" type="integer"/>
        <element minOccurs="0" name="largeVehicleRate" type="double"/>
        <element minOccurs="0" name="congestionRate" type="double"/>
        <element minOccurs="0" name="averageTravelSpeedInCongestion" type="double"/>
        <element minOccurs="0" name="observationPointName" type="string"/>
        <element minOccurs="0" name="urbanPlanType" 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"/>
        <element minOccurs="0" name="note" type="string"/>
        <element minOccurs="0" name="surveyYear" type="gYear"/>
    </sequence>
</complexType>
<complexType name="TrafficVolumePropertyType">
    <sequence>
        <element ref="uro:TrafficVolume"/>
    </sequence>
</complexType>
<group name="ValueTypeGroup">
    <choice>
        <element name="stringValue" type="string"/>
        <element name="intValue" type="integer"/>
        <element name="doubleValue" type="double"/>
        <element name="codeValue" type="gml:CodeType"/>
        <element name="measuredValue" type="gml:MeasureType"/>
        <element name="dateValue" type="date"/>
        <element name="uriValue" type="anyURI"/>
    </choice>
</group>

```

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 Listing 28 shows an example of the GML instance document that was converted from the 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.

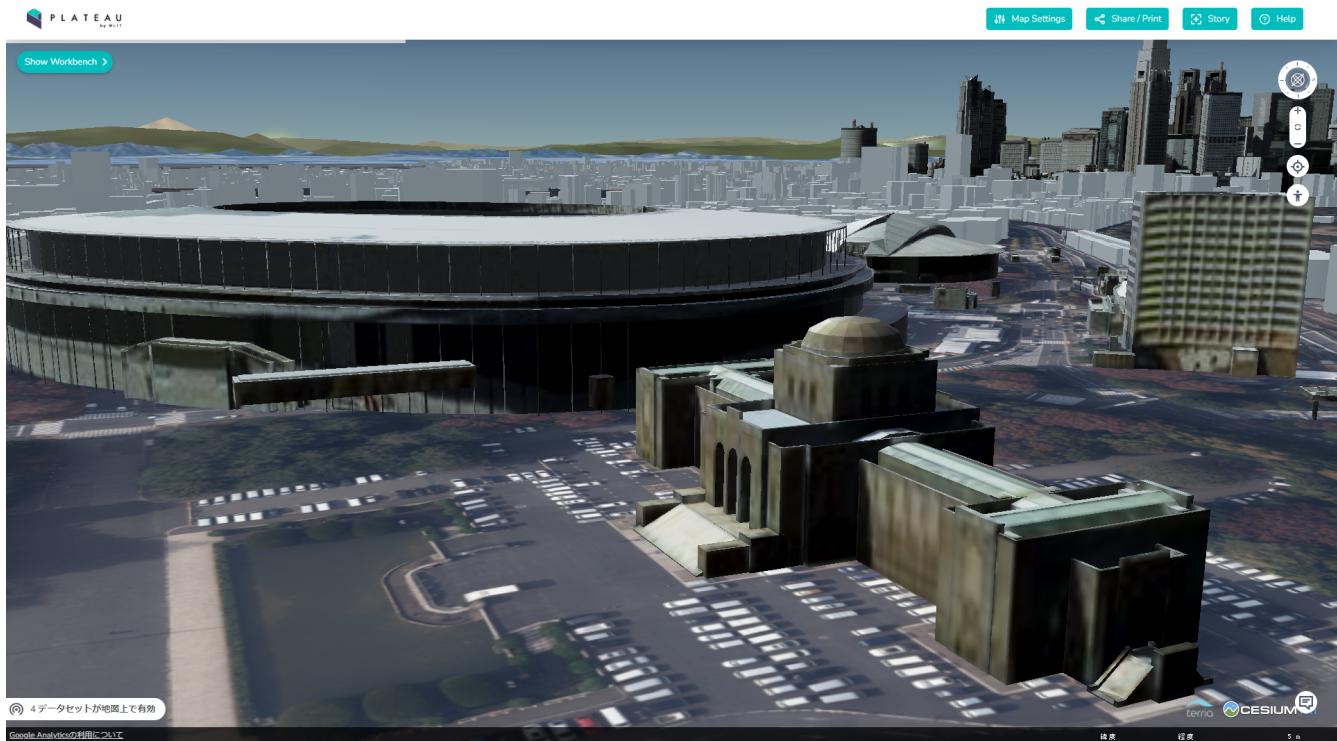


Figure 27. Meiji Memorial Picture Gallery (Mesh ID: 53394517, Building ID: 13104-bldg-53) from the Project PLATEAU

Listing 28. Example of a CityGML 3.0 GML instance document based on the converted ADE XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel
  xmlns:uro="https://www.chisou.go.jp/tiiki/toshisaisei/itoshisaisei/iur/uro/1.5" ...
  xsi:schemaLocation="https://www.chisou.go.jp/tiiki/toshisaisei/itoshisaisei/iur/uro/1.
  5 ./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: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: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: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 29](#)). The target namespace <http://www.opengis.net/citygml/3.0> is associated with this module.

Listing 29. core.xsd

```
<?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>
  <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>
  <complexType abstract="true" name="ADEOfAbstractCityObjectType">
    <sequence/>
```

```

    </complexType>
<complexType name="ADEOfAbstractCityObjectType">
    <sequence>
        <element ref="core:ADEOfAbstractCityObject"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractDynamizer"
substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractDynamizerType">
    <annotation>
        <documentation>ADEOfAbstractDynamizer acts as a hook to define properties within
an ADE that are to be added to AbstractDynamizer.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractDynamizerType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractDynamizerPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractDynamizer"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractFeature"
substitutionGroup="gml:AbstractObject" type="core:ADEOfAbstractFeatureType">
    <annotation>
        <documentation>ADEOfAbstractFeature acts as a hook to define properties within
an ADE that are to be added to AbstractFeature.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractFeatureType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractFeaturePropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractFeature"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAbstractFeatureWithLifespan"
substitutionGroup="gml:AbstractObject"
type="core:ADEOfAbstractFeatureWithLifespanType">
    <annotation>
        <documentation>ADEOfAbstractFeatureWithLifespan acts as a hook to define
properties within an ADE that are to be added to
AbstractFeatureWithLifespan.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAbstractFeatureWithLifespanType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractFeatureWithLifespanPropertyType">
    <sequence>
        <element ref="core:ADEOfAbstractFeatureWithLifespan"/>
    </sequence>

```

```

</sequence>
</complexType>
<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>
<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>

```

```

</annotation>
</element>
<complexType abstract="true" name="AbstractAppearanceType">
  <complexContent>
    <extension base="core:AbstractFeatureWithLifespanType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractAppearance"
type="core:ADEOfAbstractAppearancePropertyType">
          <annotation>
            <documentation>Augments AbstractAppearance with properties defined in an
ADE.</documentation>
          </annotation>
        </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">
  <annotation>
    <documentation>AbstractCityObject is the abstract superclass of all thematic
classes within the CityGML Conceptual Model.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractCityObjectType">
  <complexContent>
    <extension base="core:AbstractFeatureWithLifespanType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="externalReference"
type="core:ExternalReferencePropertyType">
          <annotation>
            <documentation>References external objects in other information systems
that have a relation to the city object.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="generalizesTo"
type="gml:ReferenceType">
          <annotation>
            <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>
          <appinfo>

```

```

<targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractCityObject</targetElement>
    </appinfo>
    </annotation>
</element>
<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">

```

```

<annotation>
    <documentation>Relates Dynamizer objects to the city object. These allow
timeseries data to override static attribute values of the city
object.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractCityObject"
type="core:ADEOfAbstractCityObjectPropertyType">
    <annotation>
        <documentation>Augments AbstractCityObject with properties defined in an
ADE.</documentation>
    </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractCityObjectPropertyType">
    <sequence minOccurs="0">
        <element ref="core:AbstractCityObject"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractDynamizer"
substitutionGroup="core:AbstractFeatureWithLifespan"
type="core:AbstractDynamizerType">
    <annotation>
        <documentation>AbstractDynamizer is the abstract superclass to represent
Dynamizer objects.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="AbstractDynamizerType">
    <complexContent>
        <extension base="core:AbstractFeatureWithLifespanType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractDynamizer"
type="core:ADEOfAbstractDynamizerPropertyType">
                    <annotation>
                        <documentation>Augments AbstractDynamizer with properties defined in an
ADE.</documentation>
                    </annotation>
                    </element>
            </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractDynamizer.PropertyType">
    <sequence minOccurs="0">
        <element ref="core:AbstractDynamizer"/>
    </sequence>

```

```

<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractFeature"
substitutionGroup="gml:AbstractFeature" type="core:AbstractFeatureType">
<annotation>
    <documentation>AbstractFeature is the abstract superclass of all feature types
within the CityGML Conceptual Model.</documentation>
</annotation>
</element>
<complexType abstract="true" name="AbstractFeatureType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractFeature"
type="core:ADEOfAbstractFeaturePropertyType">
                    <annotation>
                        <documentation>Augments AbstractFeature with properties defined in an
ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="AbstractFeature.PropertyType">
    <sequence minOccurs="0">
        <element ref="core:AbstractFeature"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractFeatureWithLifespan"
substitutionGroup="core:AbstractFeature" type="core:AbstractFeatureWithLifespanType">
<annotation>
    <documentation>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.</documentation>
</annotation>
</element>
<complexType abstract="true" name="AbstractFeatureWithLifespanType">
    <complexContent>
        <extension base="core:AbstractFeatureType">
            <sequence>
                <element minOccurs="0" name="creationDate" type="dateTime">
                    <annotation>
                        <documentation>Indicates the date at which a CityGML feature was added
to the CityModel.</documentation>
                    </annotation>
                </element>
                <element minOccurs="0" name="terminationDate" type="dateTime">

```

```

<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>
                <element minOccurs="0" name="lod2ImplicitRepresentation"
type="core:ImplicitGeometryPropertyType">

```

```

<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>
<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">
    <sequence minOccurs="0">
        <element ref="core:AbstractPhysicalSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractPointCloud"
substitutionGroup="core:AbstractFeature" type="core:AbstractPointCloudType">
    <annotation>
        <documentation>AbstractPointCloud is the abstract superclass to represent
PointCloud objects.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="AbstractPointCloudType">
    <complexContent>
        <extension base="core:AbstractFeatureType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractPointCloud"
type="core:ADEOfAbstractPointCloudPropertyType">
                    <annotation>

```

```

<documentation>Augments AbstractPointCloud with properties defined in an
ADE.</documentation>
    </annotation>
    </element>
    </sequence>
    </extension>
    </complexContent>
</complexType>
<complexType name="AbstractPointCloudPropertyType">
    <sequence minOccurs="0">
        <element ref="core:AbstractPointCloud"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractSpace"
substitutionGroup="core:AbstractCityObject" type="core:AbstractSpaceType">
    <annotation>
        <documentation>AbstractSpace is the abstract superclass for all types of spaces.
A space is an entity of volumetric extent in the real world.</documentation>
    </annotation>
    </element>
<complexType abstract="true" name="AbstractSpaceType">
    <complexContent>
        <extension base="core:AbstractCityObjectType">
            <sequence>
                <element minOccurs="0" name="spaceType" type="core:SpaceTypeType">
                    <annotation>
                        <documentation>Specifies the degree of openness of a
space.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="volume"
type="core:QualifiedVolumePropertyType">
                    <annotation>
                        <documentation>Specifies qualified volumes related to the
space.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="area"
type="core:QualifiedAreaPropertyType">
                    <annotation>
                        <documentation>Specifies qualified areas related to the
space.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="boundary"
type="core:AbstractSpaceBoundaryPropertyType">
                    <annotation>
                        <documentation>Relates to surfaces that bound the space.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

</element>
<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>

```

```

        </annotation>
    </element>
    <element minOccurs="0" name="lod3MultiSurface"
type="gml:MultiSurfacePropertyType">
        <annotation>
            <documentation>Relates to a 3D MultiSurface geometry that represents the
space in Level of Detail 3.</documentation>
        </annotation>
    </element>
    <element minOccurs="0" name="lod3MultiCurve"
type="gml:MultiCurvePropertyType">
        <annotation>
            <documentation>Relates to a 3D MultiCurve geometry that represents the
space in Level of Detail 3.</documentation>
        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractSpace"
type="core:ADEOfAbstractSpacePropertyType">
        <annotation>
            <documentation>Augments AbstractSpace with properties defined in an
ADE.</documentation>
        </annotation>
    </element>
    </sequence>
</extension>
</complexContent>
</complexType>
<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>
<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>
<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>
<element 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>

```

```

</element>
<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"/>

```

```

<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractVersionTransition"
substitutionGroup="core:AbstractFeatureWithLifespan"
type="core:AbstractVersionTransitionType">
    <annotation>
        <documentation>AbstractVersionTransition is the abstract superclass to represent
VersionTransition objects.</documentation>
    </annotation>
</element>
<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="AbstractVersionTransitionPropertyType">
    <sequence minOccurs="0">
        <element ref="core:AbstractVersionTransition"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
            <element ref="xAL:Address"/>
        </extension>
    </complexContent>
</complexType>

```

```

        </sequence>
    </complexType>
</element>
<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>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="Address.PropertyType">
    <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>
<complexType>
                <complexContent>
                    <extension base="gml:AbstractMetadataPropertyType">
                        <sequence minOccurs="0">

```

```

        <element ref="gml:EngineeringCRS"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>
</complexType>
</element>
<group maxOccurs="unbounded" minOccurs="0" ref="core:CityModelMemberGroup">
    <annotation>
        <documentation>Relates to all objects that are part of the CityModel.</documentation>
    </annotation>
</group>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfCityModel" type="core:ADEOfCityModelPropertyType">
    <annotation>
        <documentation>Augments the CityModel with properties defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="CityModelPropertyType">
    <sequence minOccurs="0">
        <element ref="core:CityModel"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<group name="CityModelMemberGroup">
    <annotation>
        <documentation>CityModelMember is a union type that enumerates the different types of objects that can occur as members of a city model.</documentation>
    </annotation>
    <choice>
        <element name="cityObjectMember">
            <annotation>
                <documentation>Specifies the city objects that are part of the CityModel.</documentation>
            </annotation>
</choice>
<complexType>
<complexContent>
<extension base="gml:AbstractFeatureMemberType">
    <sequence minOccurs="0">
        <element ref="core:AbstractCityObject"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</extension>
</complexContent>

```

```

    </complexType>
</element>
<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>

```

```

<documentation>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.</documentation>
</annotation>
<complexType>
  <complexContent>
    <extension base="gml:AbstractFeatureMemberType">
      <sequence minOccurs="0">
        <element ref="core:AbstractFeature"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
<element name="CityObjectRelation" substitutionGroup="gml:AbstractGML"
type="core:CityObjectRelationType">
  <annotation>
    <documentation>CityObjectRelation represents a specific relation from the city  

object in which the relation is included to another city object.</documentation>
  </annotation>
</element>
<complexType name="CityObjectRelationType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="genericAttribute"
type="core:AbstractGenericAttributePropertyType">
          <annotation>
            <documentation>Relates generic attributes to the  

CityObjectRelation.</documentation>
          </annotation>
        </element>
        <element name="relationType" type="gml:CodeType"/>
        <element name="relatedTo" type="gml:ReferenceType">
          <annotation>
            <appinfo>
              <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractCityObject</targetElement>
            </appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="CityObjectRelationPropertyType">
  <sequence minOccurs="0">
    <element ref="core:CityObjectRelation"/>

```

```

</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="ClosureSurface" substitutionGroup="core:AbstractThematicSurface"
type="core:ClosureSurfaceType">
<annotation>
    <documentation>ClosureSurface is a special type of thematic surface used to
close holes in volumetric objects. Closure surfaces are virtual (non-physical)
surfaces.</documentation>
</annotation>
</element>
<complexType name="ClosureSurfaceType">
<complexContent>
    <extension base="core:AbstractThematicSurfaceType">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="adeOfClosureSurface"
type="core:ADEOfClosureSurfacePropertyType">
                <annotation>
                    <documentation>Augments the ClosureSurface with properties defined in an
ADE.</documentation>
                </annotation>
            </element>
        </sequence>
    </extension>
</complexContent>
</complexType>
<complexType name="ClosureSurfacePropertyType">
<sequence minOccurs="0">
    <element ref="core:ClosureSurface"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<simpleType name="DoubleBetween0and1Type">
<annotation>
    <documentation>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.</documentation>
</annotation>
<restriction base="double">
    <minInclusive value="0"/>
    <maxInclusive value="1"/>
</restriction>
</simpleType>
<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>
</element>
<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="ExternalReference.PropertyType">
  <sequence>
    <element ref="core:ExternalReference"/>
  </sequence>
</complexType>
<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 re-used or referenced many times, wherever the corresponding feature occurs in the 3D city model.</documentation>
  </annotation>
</element>
<complexType name="ImplicitGeometryType">
  <complexContent>
    <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>
</extension>
</complexContent>
</complexType>
<complexType name="ImplicitGeometry.PropertyType">
    <sequence minOccurs="0">
        <element ref="core:ImplicitGeometry"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<simpleType 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="Occupancy" 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="Occupancy.PropertyType">
    <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>
    </annotation>
</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>
<simpleType name="RelativeToTerrainType">
    <annotation>
        <documentation>RelativeToTerrain enumerates the spatial relations of a city object relative to terrain in a qualitative way.</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="entirelyAboveTerrain">
            <annotation>
                <documentation>Indicates that the city object is located entirely above the

```

```


    </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>
</simpleType>
<simpleType name="RelativeToWaterType">
    <annotation>
        <documentation>RelativeToWater enumerates the spatial relations of a city object relative to the water surface in a qualitative way.</documentation>
    </annotation>
    <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>
        <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>
    </restriction>


```

```

        </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 a boundary at the bottom and on at least one side.</documentation>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>
<simpleType name="TransformationMatrix2x2Type">
    <annotation>

```

```

<documentation>TransformationMatrix2x2 is a 2 by 2 matrix represented as a list
of four double values in row major order.</documentation>
</annotation>
<restriction>
  <simpleType>
    <list itemType="double"/>
  </simpleType>
  <length value="4"/>
</restriction>
</simpleType>
<simpleType name="TransformationMatrix3x4Type">
  <annotation>
    <documentation>TransformationMatrix3x4 is a 3 by 4 matrix represented as a list
of twelve double values in row major order.</documentation>
  </annotation>
  <restriction>
    <simpleType>
      <list itemType="double"/>
    </simpleType>
    <length value="12"/>
  </restriction>
</simpleType>
<simpleType name="TransformationMatrix4x4Type">
  <annotation>
    <documentation>TransformationMatrix4x4 is a 4 by 4 matrix represented as a list
of sixteen double values in row major order.</documentation>
  </annotation>
  <restriction>
    <simpleType>
      <list itemType="double"/>
    </simpleType>
    <length value="16"/>
  </restriction>
</simpleType>
</schema>
```

C.2. Appearance

The CityGML Appearance module is defined in the XML Schema Definition file *appearance.xsd* ([Listing 30](#)). The target namespace <http://www.opengis.net/citygml/appearance/3.0> is associated with this module.

Listing 30. appearance.xsd

```

<?xml version="1.0" encoding="UTF-8"?><schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:app="http://www.opengis.net/citygml/appearance/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/appearance/3.0" version="3.0.0">
  <annotation>
```

```

<documentation>The Appearance module supports the modelling of the observable
surface properties of CityGML features in the form of textures and
material.</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="ADEOfAbstractSurfaceData"
substitutionGroup="gml:AbstractObject" type="app:ADEOfAbstractSurfaceDataType">
    <annotation>
        <documentation>ADEOfAbstractSurfaceData acts as a hook to define properties
within an ADE that are to be added to AbstractSurfaceData.</documentation>
    </annotation>
</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">
    <annotation>
        <documentation>ADEOfAbstractTexture acts as a hook to define properties within
an ADE that are to be added to AbstractTexture.</documentation>
    </annotation>
</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">
    <annotation>
        <documentation>ADEOfAppearance acts as a hook to define properties within an ADE
that are to be added to an Appearance.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAppearanceType">
    <sequence/>
</complexType>
<complexType name="ADEOfAppearancePropertyType">
    <sequence>
        <element ref="app:ADEOfAppearance"/>
    </sequence>

```

```

    </sequence>
</complexType>
<element 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 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 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 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 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>
<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>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractTexturePropertyType">
    <sequence minOccurs="0">
        <element ref="app:AbstractTexture"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element 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>
</element>
<complexType abstract="true" name="AbstractTextureParameterizationType">
    <sequence/>
</complexType>
<complexType name="AbstractTextureParameterizationPropertyType">

```

```

<sequence>
    <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>
                    <complexType>
                        <complexContent>
                            <extension base="gml:AbstractFeatureMemberType">
                                <sequence minOccurs="0">
                                    <element ref="app:AbstractSurfaceData"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                            </extension>
                        </complexContent>
                    </complexType>
                </element>
                <element maxOccurs="unbounded" 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>
<complexType name="Appearance.PropertyType">

```

```

<sequence minOccurs="0">
    <element ref="app:Appearance"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<simpleType name="ColorType">
    <annotation>
        <documentation>Color is a list of three double values between 0 and 1 defining an RGB color value.</documentation>
    </annotation>
    <restriction>
        <simpleType>
            <list itemType="core:DoubleBetween0and1Type"/>
        </simpleType>
        <length value="3"/>
    </restriction>
</simpleType>
<simpleType 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>
        <simpleType>
            <list itemType="core:DoubleBetween0and1Type"/>
        </simpleType>
        <minLength value="3"/>
        <maxLength value="4"/>
    </restriction>
</simpleType>
<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">
    <annotation>
        <documentation>Specifies the rotation and scaling of the image in form of a 2x2 matrix.</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 texture is applied.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfGeoreferencedTexture" type="app:ADEOfGeoreferencedTexturePropertyType">
    <annotation>
        <documentation>Augments the GeoreferencedTexture with properties defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="GeoreferencedTexturePropertyType">
    <sequence minOccurs="0">
        <element ref="app:GeoreferencedTexture"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="ParameterizedTexture" substitutionGroup="app:AbstractTexture" type="app:ParameterizedTextureType">
    <annotation>
        <documentation>A ParameterizedTexture is a texture that uses texture coordinates or a transformation matrix for parameterization.</documentation>
    </annotation>
</element>
<complexType name="ParameterizedTextureType">
    <complexContent>
        <extension base="app:AbstractTextureType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="textureParameterization">
                    <annotation>
                        <documentation>Relates to the texture coordinates or transformation matrices used for parameterization.</documentation>

```

```

</annotation>
<complexType>
  <sequence>
    <element ref="app:TextureAssociation"/>
  </sequence>
</complexType>
</element>
<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="TexCoordGen.PropertyType">
  <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>
<simpleType 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>
</simpleType>
<simpleType 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">
    <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>
<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>
    </sequence>
    </extension>
    </complexContent>
</complexType>
<complexType name="X3DMaterialPropertyType">
    <sequence minOccurs="0">
        <element ref="app:X3DMaterial"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>

```

C.3. Bridge

The CityGML Bridge module is defined in the XML Schema Definition file *bridge.xsd* ([Listing 31](#)). The target namespace <http://www.opengis.net/citygml/bridge/3.0> is associated with this module.

Listing 31. bridge.xsd

```

<?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 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="ADEOfBridgeConstructiveElement.PropertyType">
  <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="ADEOfBridgeFurniture.PropertyType">
  <sequence>
    <element ref="brid:ADEOfBridgeFurniture"/>
  </sequence>

```

```

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

```

```

<annotation>
    <documentation>Relates the installation objects to the Bridge or
BridgePart.</documentation>
</annotation>
<complexType>
    <complexContent>
        <extension base="gml:AbstractFeatureMemberType">
            <sequence minOccurs="0">
                <element ref="brid:BridgeInstallation"/>
            </sequence>
            <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
    </complexContent>
</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:Address.PropertyType">
    <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
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractBridge.PropertyType">
    <sequence minOccurs="0">
        <element ref="brid:AbstractBridge"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Bridge" substitutionGroup="brid:AbstractBridge"
type="brid:BridgeType">
    <annotation>
        <documentation>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]</documentation>
    </annotation>
</element>
<complexType name="BridgeType">
    <complexContent>
        <extension base="brid:AbstractBridgeType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="bridgePart">
                    <annotation>
                        <documentation>Relates the bridge parts to the Bridge.</documentation>
                    </annotation>
<complexType>
            <complexContent>
                <extension base="gml:AbstractFeatureMemberType">
                    <sequence minOccurs="0">
                        <element ref="brid:BridgePart"/>
                    </sequence>
                    <attributeGroup ref="gml:AssociationAttributeGroup"/>
                </extension>
            </complexContent>
</complexType>
        </sequence>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBridge"
type="brid:ADEOfBridgePropertyType">
    <annotation>

```

```

<documentation>Augments the Bridge with properties defined in an
ADE.</documentation>
    </annotation>
    </element>
    </sequence>
    </extension>
    </complexContent>
</complexType>
<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">
          <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>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBridgeRoom"
type="brid:ADEOfBridgeRoomPropertyType">
    <annotation>
        <documentation>Augments the BridgeRoom with properties defined in an
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="BridgeRoom.PropertyType">
    <sequence minOccurs="0">
        <element ref="brid:BridgeRoom"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>
```

C.4. Building

The CityGML Building module is defined in the XML Schema Definition file *building.xsd* ([Listing 32](#)). The target namespace <http://www.opengis.net/citygml/building/3.0> is associated with this module.

Listing 32. building.xsd

```

<?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>
<complexType abstract="true" name="ADEOfAbstractBuildingType">
  <sequence/>
</complexType>
<complexType name="ADEOfAbstractBuildingPropertyType">
  <sequence>
    <element ref="bldg:ADEOfAbstractBuilding"/>
  </sequence>
</complexType>
<element 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>
</element>
<complexType abstract="true" name="ADEOfAbstractBuildingSubdivisionType">
  <sequence/>
</complexType>
<complexType name="ADEOfAbstractBuildingSubdivisionPropertyType">
  <sequence>
    <element ref="bldg:ADEOfAbstractBuildingSubdivision"/>
  </sequence>
</complexType>
<element abstract="true" 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>
</element>
<complexType abstract="true" name="ADEOfBuildingType">
  <sequence/>
</complexType>
<complexType name="ADEOfBuildingPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuilding"/>
  </sequence>
</complexType>
<element 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>
</element>
<complexType abstract="true" name="ADEOfBuildingConstructiveElementType">
  <sequence/>
</complexType>
<complexType name="ADEOfBuildingConstructiveElementPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingConstructiveElement"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBuildingFurniture"
substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingFurnitureType">
  <annotation>
    <documentation>ADEOfBuildingFurniture acts as a hook to define properties within
an ADE that are to be added to a BuildingFurniture.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBuildingFurnitureType">
  <sequence/>
</complexType>
<complexType name="ADEOfBuildingFurniturePropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingFurniture"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBuildingInstallation"
substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingInstallationType">
  <annotation>
    <documentation>ADEOfBuildingInstallation acts as a hook to define properties
within an ADE that are to be added to a BuildingInstallation.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBuildingInstallationType">
  <sequence/>
</complexType>
<complexType name="ADEOfBuildingInstallationPropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingInstallation"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBuildingPart"
substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingPartType">
  <annotation>
    <documentation>ADEOfBuildingPart acts as a hook to define properties within an
ADE that are to be added to a BuildingPart.</documentation>
  </annotation>
</element>

```

```

<complexType abstract="true" name="ADEOfBuildingPartType">
  <sequence/>
</complexType>
<complexType name="ADEOfBuildingPart.PropertyType">
  <sequence>
    <element ref="bldg:ADEOfBuildingPart"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfBuildingRoom"
substitutionGroup="gml:AbstractObject" type="bldg:ADEOfBuildingRoomType">
  <annotation>
    <documentation>ADEOfBuildingRoom acts as a hook to define properties within an
ADE that are to be added to a BuildingRoom.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfBuildingRoomType">
  <sequence/>
</complexType>
<complexType name="ADEOfBuildingRoom.PropertyType">
  <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="ADEOfBuildingUnit.PropertyType">
  <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="ADEOfStorey.PropertyType">
  <sequence>
    <element ref="bldg:ADEOfStorey"/>
  </sequence>

```

```

    </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
BuildingPart.</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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

</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>
<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>
            <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:Address.PropertyType">
    <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
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractBuildingPropertyType">
    <sequence minOccurs="0">
        <element ref="bldg:AbstractBuilding"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractBuildingSubdivision"
substitutionGroup="core:AbstractLogicalSpace"
type="bldg:AbstractBuildingSubdivisionType">
    <annotation>
        <documentation>AbstractBuildingSubdivision is the abstract superclass for
different kinds of logical building subdivisions.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="AbstractBuildingSubdivisionType">
    <complexContent>
        <extension base="core:AbstractLogicalSpaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the building
subdivision.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the building
subdivision.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the building
subdivision.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="elevation"
type="con:ElevationPropertyType">
        <annotation>
            <documentation>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]</documentation>
        </annotation>
    </element>
    <element minOccurs="0" name="sortKey" type="double">
        <annotation>
            <documentation>Defines an order among the objects that belong to the
building subdivision. An example is the sorting of storeys.</documentation>
        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0"
name="buildingConstructiveElement"
type="bldg:BuildingConstructiveElementPropertyType">
        <annotation>
            <documentation>Relates the constructive elements to the building
subdivision.</documentation>
        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="buildingFurniture">
        <annotation>
            <documentation>Relates the furniture objects to the building
subdivision.</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="buildingInstallation">
        <annotation>
            <documentation>Relates the installation objects to the building
subdivision.</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"
type="bldg:BuildingRoomPropertyType">
    <annotation>
        <documentation>Relates the rooms to the building
subdivision.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfAbstractBuildingSubdivision"
type="bldg:ADEOfAbstractBuildingSubdivisionPropertyType">
    <annotation>
        <documentation>Augments AbstractBuildingSubdivision with properties
defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<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>
        </extension>
    </complexContent>
</complexType>

```

```

<complexContent>
    <extension base="gml:AbstractFeatureMemberType">
        <sequence minOccurs="0">
            <element ref="bldg:BuildingPart"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
</complexContent>
</complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfBuilding"
type="bldg:ADEOfBuildingPropertyType">
    <annotation>
        <documentation>Augments the Building with properties defined in an
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="Building.PropertyType">
    <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>
    </annotation>
</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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
        <annotation>
            <documentation>Specifies the actual uses of the
BuildingFurniture.</documentation>
        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingFurniture"
type="bldg:ADEOfBuildingFurniturePropertyType">
        <annotation>
            <documentation>Augments the BuildingFurniture with properties defined in
an ADE.</documentation>
        </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="BuildingFurniturePropertyType">
    <sequence minOccurs="0">
        <element ref="bldg:BuildingFurniture"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
<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">
    <sequence minOccurs="0">
        <element ref="bldg:BuildingInstallation"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="BuildingPart" substitutionGroup="bldg:AbstractBuilding"
type="bldg:BuildingPartType">
    <annotation>
        <documentation>A BuildingPart is a physical or functional subdivision of a
Building. It would be considered a Building, if it were not part of a collection of
other BuildingParts.</documentation>
    </annotation>
</element>
<complexType name="BuildingPartType">
    <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>
        </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">
<annotation>
    <documentation>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).</documentation>
</annotation>
</element>
<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>

```

```

<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="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>
</sequence>
</extension>
</complexContent>
</complexType>
<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>

```

```

</annotation>
</element>
<complexType name="BuildingUnitType">
  <complexContent>
    <extension base="bldg:AbstractBuildingSubdivisionType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="storey"
type="bldg:Storey.PropertyType">
          <annotation>
            <documentation>Relates to the storeys on which the BuildingUnit is
located.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="address"
type="core:Address.PropertyType">
          <annotation>
            <documentation>Relates to the addresses that are assigned to the
BuildingUnit.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfBuildingUnit"
type="bldg:ADEOfBuildingUnit.PropertyType">
          <annotation>
            <documentation>Augments the BuildingUnit with properties defined in an
ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BuildingUnit.PropertyType">
  <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">
  <annotation>
    <documentation>The RoomHeight represents a vertical distance (measured or
estimated) between a low reference and a high reference. [cf. INSPIRE]</documentation>
  </annotation>
</element>
<complexType name="RoomHeightType">
  <sequence>
    <element name="highReference" type="gml:CodeType">
      <annotation>
        <documentation>Indicates the high point used to calculate the value of the
room height.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

```

```

        </annotation>
    </element>
    <element name="lowReference" type="gml:CodeType">
        <annotation>
            <documentation>Indicates the low point used to calculate the value of the room height.</documentation>
        </annotation>
    </element>
    <element name="status" type="con:HeightStatusValueType">
        <annotation>
            <documentation>Indicates the way the room height has been captured.</documentation>
        </annotation>
    </element>
    <element name="value" type="gml:LengthType">
        <annotation>
            <documentation>Specifies the value of the room height.</documentation>
        </annotation>
    </element>
</sequence>
</complexType>
<complexType name="RoomHeightPropertyType">
    <sequence>
        <element ref="bldg:RoomHeight"/>
    </sequence>
</complexType>
<element name="Storey" substitutionGroup="bldg:AbstractBuildingSubdivision" type="bldg:StoreyType">
    <annotation>
        <documentation>A Storey is typically a horizontal section of a Building. Storeys are not always defined according to the building structure, but can also be defined according to logical considerations.</documentation>
    </annotation>
</element>
<complexType name="StoreyType">
    <complexContent>
        <extension base="bldg:AbstractBuildingSubdivisionType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="buildingUnit" type="bldg:BuildingUnit.PropertyType">
                    <annotation>
                        <documentation>Relates to the building units that belong to the Storey.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfStorey" type="bldg:ADEOfStorey.PropertyType">
                    <annotation>
                        <documentation>Augments the Storey with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </element>
    </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="StoreyPropertyType">
    <sequence minOccurs="0">
        <element ref="bldg:Storey"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>

```

C.5. City Furniture

The CityGML City Furniture module is defined in the XML Schema Definition file *cityFurniture.xsd* ([Listing 33](#)). The target namespace <http://www.opengis.net/citygml/cityfurniture/3.0> is associated with this module.

Listing 33. cityFurniture.xsd

```

<?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>
    </annotation>
  </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 `cityObjectGroup.xsd` ([Listing 34](#)). The target namespace <http://www.opengis.net/citygml/cityobjectgroup/3.0> is associated with this module.

Listing 34. cityObjectGroup.xsd

```
<?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">
      <sequence/>
    </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>
</schema>
```

```

</annotation>
</element>
<complexType name="CityObjectGroupType">
  <complexContent>
    <extension base="core:AbstractLogicalSpaceType">
      <sequence>
        <element minOccurs="0" name="class" type="gml:CodeType">
          <annotation>
            <documentation>Indicates the specific type of the
CityObjectGroup.</documentation>
          </annotation>
        </element>
        <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>
          <appinfo>
            <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractCityObject</targetElement>
          </appinfo>
        </annotation>
      </sequence>
      <element maxOccurs="unbounded" minOccurs="0" name="adeOfCityObjectGroup"
type="grp:ADEOfCityObjectGroupPropertyType">
        <annotation>
          <documentation>Augments the CityObjectGroup with properties defined in

```

```

an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="CityObjectGroupPropertyType">
    <sequence minOccurs="0">
        <element ref="grp:CityObjectGroup"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Role" substitutionGroup="gml:AbstractGML" type="grp:RoleType">
    <annotation>
        <documentation>Role qualifies the function of a city object within the
CityObjectGroup.</documentation>
    </annotation>
</element>
<complexType name="RoleType">
    <complexContent>
        <extension base="gml:AbstractGMLType">
            <sequence>
                <element minOccurs="0" name="role" type="string"/>
                <element name="groupMember">
                    <annotation>
                        <appinfo>
                            <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractCityObject</targetElement>
                        </appinfo>
                    </annotation>
                    <complexType>
                        <complexContent>
                            <extension base="gml:AbstractFeatureMemberType">
                                <sequence/>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                            </extension>
                        </complexContent>
                    </complexType>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="Role.PropertyType">
    <sequence minOccurs="0">
        <element ref="grp:Role"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>

```

```
</complexType>
</schema>
```

C.7. Construction

The CityGML Construction module is defined in the XML Schema Definition file *construction.xsd* ([Listing 35](#)). The target namespace <http://www.opengis.net/citygml/construction/3.0> is associated with this module.

Listing 35. construction.xsd

```
<?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>
<element 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>
<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>
<element abstract="true" name="ADEOfDoorSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfDoorSurfaceType">
  <annotation>
    <documentation>ADEOfDoorSurface acts as a hook to define properties within an
ADE that are to be added to a DoorSurface.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfDoorSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfDoorSurface.PropertyType">
  <sequence>
    <element ref="con:ADEOfDoorSurface"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfFloorSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfFloorSurfaceType">
  <annotation>
    <documentation>ADEOfFloorSurface acts as a hook to define properties within an
ADE that are to be added to a FloorSurface.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfFloorSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfFloorSurface.PropertyType">
  <sequence>
    <element ref="con:ADEOfFloorSurface"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfGroundSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfGroundSurfaceType">
  <annotation>
    <documentation>ADEOfGroundSurface acts as a hook to define properties within an
ADE that are to be added to a GroundSurface.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfGroundSurfaceType">

```

```

<sequence/>
</complexType>
<complexType name="ADEOfGroundSurfacePropertyType">
  <sequence>
    <element ref="con:ADEOfGroundSurface"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfInteriorWallSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfInteriorWallSurfaceType">
  <annotation>
    <documentation>ADEOfInteriorWallSurface acts as a hook to define properties
within an ADE that are to be added to an InteriorWallSurface.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfInteriorWallSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfInteriorWallSurfacePropertyType">
  <sequence>
    <element ref="con:ADEOfInteriorWallSurface"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfOtherConstruction"
substitutionGroup="gml:AbstractObject" type="con:ADEOfOtherConstructionType">
  <annotation>
    <documentation>ADEOfOtherConstruction acts as a hook to define properties within
an ADE that are to be added to an OtherConstruction.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfOtherConstructionType">
  <sequence/>
</complexType>
<complexType name="ADEOfOtherConstructionPropertyType">
  <sequence>
    <element ref="con:ADEOfOtherConstruction"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfOuterCeilingSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfOuterCeilingSurfaceType">
  <annotation>
    <documentation>ADEOfOuterCeilingSurface acts as a hook to define properties
within an ADE that are to be added to an OuterCeilingSurface.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfOuterCeilingSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfOuterCeilingSurfacePropertyType">
  <sequence>
    <element ref="con:ADEOfOuterCeilingSurface"/>
  </sequence>
</complexType>

```

```

    </complexType>
    <element abstract="true" name="ADEOfOuterFloorSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfOuterFloorSurfaceType">
        <annotation>
            <documentation>ADEOfOuterFloorSurface acts as a hook to define properties within
an ADE that are to be added to an OuterFloorSurface.</documentation>
        </annotation>
    </element>
    <complexType abstract="true" name="ADEOfOuterFloorSurfaceType">
        <sequence/>
    </complexType>
    <complexType name="ADEOfOuterFloorSurfacePropertyType">
        <sequence>
            <element ref="con:ADEOfOuterFloorSurface"/>
        </sequence>
    </complexType>
    <element abstract="true" name="ADEOfRoofSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfRoofSurfaceType">
        <annotation>
            <documentation>ADEOfRoofSurface acts as a hook to define properties within an
ADE that are to be added to a RoofSurface.</documentation>
        </annotation>
    </element>
    <complexType abstract="true" name="ADEOfRoofSurfaceType">
        <sequence/>
    </complexType>
    <complexType name="ADEOfRoofSurfacePropertyType">
        <sequence>
            <element ref="con:ADEOfRoofSurface"/>
        </sequence>
    </complexType>
    <element abstract="true" name="ADEOfWallSurface"
substitutionGroup="gml:AbstractObject" type="con:ADEOfWallSurfaceType">
        <annotation>
            <documentation>ADEOfWallSurface acts as a hook to define properties within an
ADE that are to be added to a WallSurface.</documentation>
        </annotation>
    </element>
    <complexType abstract="true" name="ADEOfWallSurfaceType">
        <sequence/>
    </complexType>
    <complexType name="ADEOfWallSurfacePropertyType">
        <sequence>
            <element ref="con:ADEOfWallSurface"/>
        </sequence>
    </complexType>
    <element abstract="true" name="ADEOfWindow" substitutionGroup="gml:AbstractObject"
type="con:ADEOfWindowType">
        <annotation>
            <documentation>ADEOfWindow acts as a hook to define properties within an ADE
that are to be added to a Window.</documentation>
        </annotation>

```

```

</annotation>
</element>
<complexType abstract="true" name="ADEOfWindowType">
  <sequence/>
</complexType>
<complexType name="ADEOfWindow.PropertyType">
  <sequence>
    <element ref="con:ADEOfWindow"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfWindowSurface"
substitutionGroup="gml: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="ADEOfWindowSurface.PropertyType">
  <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>
    <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>
</extension>
</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">
<annotation>
<documentation>AbstractConstructionSurface is the abstract superclass for
different kinds of surfaces that bound a construction.</documentation>
</annotation>
</element>
<complexType abstract="true" name="AbstractConstructionSurfaceType">
<complexContent>
<extension base="core:AbstractThematicSurfaceType">
<sequence>
<element maxOccurs="unbounded" minOccurs="0" name="fillingSurface"
type="con:AbstractFillingSurfacePropertyType">
<annotation>
<documentation>Relates to the surfaces that seal the openings of the
construction surface.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfAbstractConstructionSurface"
type="con:ADEOfAbstractConstructionSurfacePropertyType">
<annotation>
<documentation>Augments AbstractConstructionSurface with properties
defined in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractConstructionSurfacePropertyType">
<sequence minOccurs="0">
<element ref="con:AbstractConstructionSurface"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractConstructiveElement"
substitutionGroup="core:AbstractOccupiedSpace"
type="con:AbstractConstructiveElementType">
<annotation>
<documentation>AbstractConstructiveElement is the abstract superclass for the
representation of volumetric elements of a construction. Examples are walls, beams,
slabs.</documentation>
</annotation>
</element>

```

```

<complexType abstract="true" name="AbstractConstructiveElementType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element minOccurs="0" name="isStructuralElement" type="boolean">
          <annotation>
            <documentation>Indicates whether the constructive element is essential from a structural point of view. A structural element cannot be omitted without collapsing of the construction. Examples are pylons and anchorages of bridges.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="filling" type="con:AbstractFillingElementPropertyType">
          <annotation>
            <documentation>Relates to the elements that fill the opening of the constructive element.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractConstructiveElement" type="con:ADEOfAbstractConstructiveElementPropertyType">
          <annotation>
            <documentation>Augments AbstractConstructiveElement with properties defined in an ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AbstractConstructiveElementPropertyType">
  <sequence minOccurs="0">
    <element ref="con:AbstractConstructiveElement"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractFillingElement" substitutionGroup="core:AbstractOccupiedSpace" type="con:AbstractFillingElementType">
  <annotation>
    <documentation>AbstractFillingElement is the abstract superclass for different kinds of elements that fill the openings of a construction.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="AbstractFillingElementType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractFillingElement" type="con:ADEOfAbstractFillingElementPropertyType">

```

```

<annotation>
    <documentation>Augments AbstractFillingElement with properties defined
in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractFillingElementType">
    <sequence minOccurs="0">
        <element ref="con:AbstractFillingElement"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractFillingSurface"
substitutionGroup="core:AbstractThematicSurface"
type="con:AbstractFillingSurfaceType">
    <annotation>
        <documentation>AbstractFillingSurface is the abstract superclass for different
kinds of surfaces that seal openings filled by filling elements.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="AbstractFillingSurfaceType">
    <complexContent>
        <extension base="core:AbstractThematicSurfaceType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0"
name="adeOfAbstractFillingSurface" type="con:ADEOfAbstractFillingSurfacePropertyType">
                    <annotation>
                        <documentation>Augments AbstractFillingSurface with properties defined
in an ADE.</documentation>
                    </annotation>
</element>
            </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AbstractFillingSurfacePropertyType">
    <sequence minOccurs="0">
        <element ref="con:AbstractFillingSurface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AbstractFurniture"
substitutionGroup="core:AbstractOccupiedSpace" type="con:AbstractFurnitureType">
    <annotation>
        <documentation>AbstractFurniture is the abstract superclass for the
representation of furniture objects of a construction.</documentation>
    </annotation>

```

```

</annotation>
</element>
<complexType abstract="true" name="AbstractFurnitureType">
  <complexContent>
    <extension base="core:AbstractOccupiedSpaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractFurniture"
type="con:ADEOfAbstractFurniturePropertyType">
          <annotation>
            <documentation>Augments AbstractFurniture with properties defined in an
ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</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>

```

```

        </complexContent>
    </complexType>
    <complexType name="AbstractInstallationPropertyType">
        <sequence minOccurs="0">
            <element ref="con:AbstractInstallation"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
    <element name="CeilingSurface" substitutionGroup="con:AbstractConstructionSurface"
type="con:CeilingSurfaceType">
        <annotation>
            <documentation>A CeilingSurface is a surface that represents the interior
ceiling of a construction. An example is the ceiling of a room.</documentation>
        </annotation>
    </element>
    <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>
        </restriction>
    </simpleType>

```

```

<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">
    <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="Door.PropertyType">
    <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:Address.PropertyType">
                    <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="DoorSurface.PropertyType">
    <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>
<documentation>Elevation is a data type that includes the elevation value itself and information on how this elevation was measured. [cf. INSPIRE]</documentation>
</annotation>
</element>
<complexType name="ElevationType">
<sequence>
<element name="elevationReference" type="gml:CodeType">
<annotation>
<documentation>Specifies the level from which the elevation was measured. [cf. INSPIRE]</documentation>
</annotation>
</element>
<element name="elevationValue" type="gml:DirectPositionType">
<annotation>
<documentation>Specifies the value of the elevation. [cf. INSPIRE]</documentation>
</annotation>
</element>
</sequence>
</complexType>
<complexType name="ElevationPropertyType">
<sequence>
<element ref="con:Elevation"/>
</sequence>
</complexType>
<element name="FloorSurface" substitutionGroup="con:AbstractConstructionSurface"
type="con:FloorSurfaceType">
<annotation>
<documentation>A FloorSurface is surface that represents the interior floor of a construction. An example is the floor of a room.</documentation>
</annotation>
</element>
<complexType name="FloorSurfaceType">
<complexContent>
<extension base="con:AbstractConstructionSurfaceType">
<sequence>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfFloorSurface"
type="con:ADEOfFloorSurfacePropertyType">
<annotation>
<documentation>Augments the FloorSurface with properties defined in an ADE.</documentation>
</annotation>
</element>
</sequence>

```

```

</extension>
</complexContent>
</complexType>
<complexType name="FloorSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:FloorSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="GroundSurface" substitutionGroup="con:AbstractConstructionSurface"
type="con:GroundSurfaceType">
  <annotation>
    <documentation>A GroundSurface is a surface that represents the ground plate of
a construction. The polygon defining the ground plate is congruent with the footprint
of the construction.</documentation>
  </annotation>
</element>
<complexType name="GroundSurfaceType">
  <complexContent>
    <extension base="con:AbstractConstructionSurfaceType">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfGroundSurface"
type="con:ADEOfGroundSurfacePropertyType">
          <annotation>
            <documentation>Augments the GroundSurface with properties defined in an
ADE.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="GroundSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="con:GroundSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
<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>
<complexType name="Height.PropertyType">
    <sequence>
        <element ref="con:Height"/>
    </sequence>
</complexType>
<simpleType 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>
</simpleType>
<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>
        <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>
    </complexContent>
</complexType>
<complexType name="InteriorWallSurface.PropertyType">
    <sequence minOccurs="0">
        <element ref="con:InteriorWallSurface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

<element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
    <annotation>
        <documentation>Specifies the actual uses of the
OtherConstruction.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfOtherConstruction"
type="con:ADEOfOtherConstructionPropertyType">
    <annotation>
        <documentation>Augments the OtherConstruction with properties defined in
an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="OtherConstructionPropertyType">
    <sequence minOccurs="0">
        <element ref="con:OtherConstruction"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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 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>
</extension>
</complexContent>
</complexType>
<complexType name="OuterCeilingSurfacePropertyType">
    <sequence minOccurs="0">

```

```

<element ref="con:OuterCeilingSurface"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="OuterFloorSurface"
substitutionGroup="con:AbstractConstructionSurface" type="con:OuterFloorSurfaceType">
<annotation>
    <documentation>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.</documentation>
</annotation>
</element>
<complexType name="OuterFloorSurfaceType">
    <complexContent>
        <extension base="con:AbstractConstructionSurfaceType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfOuterFloorSurface"
type="con:ADEOfOuterFloorSurfacePropertyType">
                    <annotation>
                        <documentation>Augments the OuterFloorSurface with properties defined in
an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="OuterFloorSurfacePropertyType">
    <sequence minOccurs="0">
        <element ref="con:OuterFloorSurface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<simpleType name="RelationToConstructionType">
    <annotation>
        <documentation>RelationToConstruction is an enumeration used to describe whether
an installation is positioned inside and/or outside of a construction.</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="inside">
            <annotation>
                <documentation>Indicates that the installation is positioned inside of the
construction.</documentation>
            </annotation>
        </enumeration>
        <enumeration value="outside">
            <annotation>
                <documentation>Indicates that the installation is positioned outside of the
construction.</documentation>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>

```

```

        </annotation>
    </enumeration>
    <enumeration value="bothInsideAndOutside">
        <annotation>
            <documentation>Indicates that the installation is positioned inside as well as outside of the construction.</documentation>
        </annotation>
    </enumeration>
</restriction>
</simpleType>
<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:ADEOfWindow.PropertyType">
                    <annotation>

```

```

<documentation>Augments the Window with properties defined in an
ADE.</documentation>
    </annotation>
    </element>
    </sequence>
    </extension>
    </complexContent>
</complexType>
<complexType name="Window.PropertyType">
    <sequence minOccurs="0">
        <element ref="con:Window"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="WindowSurface" substitutionGroup="con:AbstractFillingSurface"
type="con:WindowSurfaceType">
    <annotation>
        <documentation>A WindowSurface is either a boundary surface of a Window feature
or a surface that seals an opening filled by a window.</documentation>
    </annotation>
</element>
<complexType name="WindowSurfaceType">
    <complexContent>
        <extension base="con:AbstractFillingSurfaceType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfWindowSurface"
type="con:ADEOfWindowSurface.PropertyType">
                    <annotation>
                        <documentation>Augments the WindowSurface with properties defined in an
ADE.</documentation>
                    </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="WindowSurface.PropertyType">
        <sequence minOccurs="0">
            <element ref="con:WindowSurface"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
</schema>
```

C.8. Dynamizer

The CityGML Dynamizer module is defined in the XML Schema Definition file *dynamizer.xsd* ([Listing 36](#)). The target namespace <http://www.opengis.net/citygml/dynamizer/3.0> is associated with

this module.

Listing 36. dynamizer.xsd

```
<?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"
  xmlns:gml="http://www.opengis.net/gml/3.2" elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/dynamizer/3.0" version="3.0.0">
  <annotation>
    <documentation>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.</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="ADEOfAbstractAtomicTimeseries"
  substitutionGroup="gml:AbstractObject" type="dyn:ADEOfAbstractAtomicTimeseriesType">
    <annotation>
      <documentation>ADEOfAbstractAtomicTimeseries acts as a hook to define properties
      within an ADE that are to be added to AbstractAtomicTimeseries.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAbstractAtomicTimeseriesType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractAtomicTimeseriesPropertyType">
    <sequence>
      <element ref="dyn:ADEOfAbstractAtomicTimeseries"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfAbstractTimeseries"
  substitutionGroup="gml:AbstractObject" type="dyn:ADEOfAbstractTimeseriesType">
    <annotation>
      <documentation>ADEOfAbstractTimeseries acts as a hook to define properties
      within an ADE that are to be added to AbstractTimeseries.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAbstractTimeseriesType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractTimeseriesPropertyType">
    <sequence>
      <element ref="dyn:ADEOfAbstractTimeseries"/>
    </sequence>
  </complexType>
```

```

<element abstract="true" name="ADEOfCompositeTimeseries"
substitutionGroup="gml:AbstractObject" type="dyn:ADEOfCompositeTimeseriesType">
  <annotation>
    <documentation>ADEOfCompositeTimeseries acts as a hook to define properties
within an ADE that are to be added to a CompositeTimeseries.</documentation>
  </annotation>
</element>
<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>
<element 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>
</element>
<complexType abstract="true" name="ADEOfTabulatedFileTimeseriesType">
  <sequence/>
</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 maxOccurs="unbounded" minOccurs="0" name="obs">
          <annotation>
            <documentation>Observation values for the atomic timeseries.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

name="adeOfAbstractAtomicTimeseries"
type="dyn:ADEOfAbstractAtomicTimeseriesPropertyType">
    <annotation>
        <documentation>Augments AbstractAtomicTimeseries with properties defined
in an ADE.</documentation>
    </annotation>
    </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>
<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>

```

```

<complexType name="AbstractTimeseriesPropertyType">
    <sequence minOccurs="0">
        <element ref="dyn:AbstractTimeseries"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="CompositeTimeseries" substitutionGroup="dyn:AbstractTimeseries"
type="dyn:CompositeTimeseriesType">
    <annotation>
        <documentation>A CompositeTimeseries is a (possibly recursive) aggregation of
atomic and composite timeseries. The components of a composite timeseries must have
non-overlapping time intervals.</documentation>
    </annotation>
</element>
<complexType name="CompositeTimeseriesType">
    <complexContent>
        <extension base="dyn:AbstractTimeseriesType">
            <sequence>
                <element maxOccurs="unbounded" name="component"
type="dyn:TimeseriesComponentPropertyType">
                    <annotation>
                        <documentation>Relates to the atomic and composite timeseries that are
part of the CompositeTimeseries. The referenced timeseries are sequentially
ordered.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfCompositeTimeseries"
type="dyn:ADEOfCompositeTimeseriesPropertyType">
                    <annotation>
                        <documentation>Augments the CompositeTimeseries with properties defined
in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="CompositeTimeseriesPropertyType">
    <sequence minOccurs="0">
        <element ref="dyn:CompositeTimeseries"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Dynamizer" substitutionGroup="core:AbstractDynamizer"
type="dyn:DynamizerType">
    <annotation>
        <documentation>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.</documentation>
    </annotation>
</element>
<complexType name="DynamizerType">
    <complexContent>
        <extension base="core:AbstractDynamizerType">
            <sequence>
                <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>
                </element>
                <complexType>
                    <complexContent>
                        <extension base="gml:AbstractFeatureMemberType">
                            <sequence minOccurs="0">
                                <element ref="dyn:AbstractTimeseries"/>
                            </sequence>
                            <attributeGroup ref="gml:AssociationAttributeGroup"/>
                        </extension>
                    </complexContent>
                </complexType>
            </sequence>
        </extension>
    </complexContent>
<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:DynamizerType">
    <annotation>
        <documentation>A collection of dynamizers that are applied to a feature type. The order of the elements defines the order of application of the dynamizers. The first dynamizer in the list is applied to all features of the feature type. Subsequent dynamizers are applied to features that have not been modified by previous dynamizers in the list. This allows for complex logic to be applied to features based on their history of modification by previous dynamizers.</documentation>
    </annotation>

```

```

type="dyn:ADEOfDynamizer.PropertyType">
    <annotation>
        <documentation>Augments the Dynamizer with properties defined in an ADE.</documentation>
    </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="Dynamizer.PropertyType">
    <sequence minOccurs="0">
        <element ref="dyn:Dynamizer"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="GenericTimeseries" substitutionGroup="dyn:AbstractAtomicTimeseries" type="dyn:GenericTimeseriesType">
    <annotation>
        <documentation>A GenericTimeseries represents time-varying data in the form of embedded time-value-pairs of a specific data type for a single contiguous time interval.</documentation>
    </annotation>
    </element>
<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:ADEOfGenericTimeseries.PropertyType">
                    <annotation>
                        <documentation>Augments the GenericTimeseries with properties defined in an ADE.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

    </complexContent>
</complexType>
<complexType name="GenericTimeseriesPropertyType">
    <sequence minOccurs="0">
        <element ref="dyn:GenericTimeseries"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
<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>
    <appinfo>
        <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractCityObject</targetElement>
    </appinfo>
</annotation>
</element>
</sequence>
</complexType>
<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 & 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 maxOccurs="unbounded" 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>
<complexType name="StandardFileTimeseriesPropertyType">
    <sequence minOccurs="0">
        <element ref="dyn:StandardFileTimeseries"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TabulatedFileTimeseries"
substitutionGroup="dyn:AbstractAtomicTimeseries"
type="dyn:TabulatedFileTimeseriesType">
    <annotation>
        <documentation>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.</documentation>
    </annotation>
</element>
<complexType name="TabulatedFileTimeseriesType">
    <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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </annotation>
    </element>
    <element name="valueType" type="dyn:TimeseriesTypeValueType">
        <annotation>
            <documentation>Indicates the specific type of the timeseries
data.</documentation>
        </annotation>
    </element>
    <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>
        <documentation>Specifies the number of the column that stores the value
of the time-value-pair.</documentation>
    </annotation>
</element>
<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>
<element minOccurs="0" name="doubleValue" type="double">
    <annotation>
        <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>
<complexType>
    <complexContent>
        <extension base="gml:AbstractMemberType">
            <sequence minOccurs="0">
                <element ref="core:ImplicitGeometry"/>
            </sequence>
            <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
    </complexContent>
</complexType>

```

```

        </extension>
    </complexContent>
</complexType>
</element>
<element minOccurs="0" name="appearanceValue">
    <annotation>
        <documentation>Specifies the "Appearance" value of the TimeValuePair.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="core:AbstractAppearance"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>
</sequence>
</complexType>
<complexType name="TimeValuePair.PropertyType">
    <sequence>
        <element ref="dyn:TimeValuePair"/>
    </sequence>
</complexType>
<element name="TimeseriesComponent" substitutionGroup="gml:AbstractObject" type="dyn:TimeseriesComponentType">
    <annotation>
        <documentation>TimeseriesComponent represents an element of a CompositeTimeseries.</documentation>
    </annotation>
</element>
<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="TimeseriesComponent.PropertyType">
    <sequence>
        <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 geometries.</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>
<enumeration value="appearance">
    <annotation>
        <documentation>Indicates that the values of the GenericTimeseries are of type "Appearance".</documentation>
    </annotation>
</enumeration>
</restriction>
</simpleType>
</schema>

```

C.9. Generics

The CityGML Generics module is defined in the XML Schema Definition file *generics.xsd* ([Listing 37](#)). The target namespace <http://www.opengis.net/citygml/generics/3.0> is associated with this module.

Listing 37. generics.xsd

```

<?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>ADEOfGenericOccupiedSpace acts as a hook to define properties
within an ADE that are to be added to a GenericOccupiedSpace.</documentation>
  </annotation>
</element>
<complexType abstract="true" name="ADEOfGenericOccupiedSpaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfGenericOccupiedSpace.PropertyType">
  <sequence>
    <element ref="gen:ADEOfGenericOccupiedSpace"/>
  </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 abstract="true" name="ADEOfGenericThematicSurfaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfGenericThematicSurface.PropertyType">
  <sequence>
    <element ref="gen:ADEOfGenericThematicSurface"/>
  </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 abstract="true" name="ADEOfGenericUnoccupiedSpaceType">
  <sequence/>
</complexType>
<complexType name="ADEOfGenericUnoccupiedSpace.PropertyType">
  <sequence>
    <element ref="gen:ADEOfGenericUnoccupiedSpace"/>
  </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 name="CodeAttributeType">
  <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>
<complexType name="CodeAttributePropertyType">
  <sequence>
    <element ref="gen:CodeAttribute"/>
  </sequence>
</complexType>
<element name="DateAttribute" substitutionGroup="core:AbstractGenericAttribute" type="gen:DateAttributeType">
  <annotation>
    <documentation>DateAttribute is a data type used to define generic attributes of type "Date".</documentation>
  </annotation>
</element>
<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>

```

```

        </extension>
    </complexContent>
</complexType>
<complexType name="DateAttributePropertyType">
    <sequence>
        <element ref="gen:DateAttribute"/>
    </sequence>
</complexType>
<element name="DoubleAttribute" substitutionGroup="core:AbstractGenericAttribute"
type="gen:DoubleAttributeType">
    <annotation>
        <documentation>DoubleAttribute is a data type used to define generic attributes
of type "Double".</documentation>
    </annotation>
</element>
<complexType name="DoubleAttributeType">
    <complexContent>
        <extension base="core:AbstractGenericAttributeType">
            <sequence>
                <element name="name" type="string">
                    <annotation>
                        <documentation>Specifies the name of the
DoubleAttribute.</documentation>
                    </annotation>
                </element>
                <element name="value" type="double">
                    <annotation>
                        <documentation>Specifies the "Double" value.</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</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>Associates the GenericAttributeSet with an authority that
maintains the collection of generic attributes.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" name="genericAttribute"
type="core:AbstractGenericAttributePropertyType">
    <annotation>
        <documentation>Relates to the generic attributes that are part of the
GenericAttributeSet.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="GenericAttributeSet.PropertyType">
    <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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
            <annotation>
                <documentation>Specifies the actual uses of the
GenericLogicalSpace.</documentation>
            </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="adeOfGenericLogicalSpace"
type="gen:ADEOfGenericLogicalSpacePropertyType">
            <annotation>
                <documentation>Augments the GenericLogicalSpace with properties defined
in an ADE.</documentation>
            </annotation>
        </element>
    </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="GenericLogicalSpacePropertyType">
    <sequence minOccurs="0">
        <element ref="gen:GenericLogicalSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfGenericOccupiedSpace" type="gen:ADEOfGenericOccupiedSpacePropertyType">
    <annotation>
        <documentation>Augments the GenericOccupiedSpace with properties defined
in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="GenericOccupiedSpacePropertyType">
    <sequence minOccurs="0">
        <element ref="gen:GenericOccupiedSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
<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>Augments the GenericThematicSurface with properties
defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="GenericThematicSurfacePropertyType">
    <sequence minOccurs="0">
        <element ref="gen:GenericThematicSurface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="GenericUnoccupiedSpace"
substitutionGroup="core:AbstractUnoccupiedSpace"
type="gen:GenericUnoccupiedSpaceType">
    <annotation>
        <documentation>A GenericUnoccupiedSpace is a space that is not represented by
any explicitly modelled AbstractUnoccupiedSpace subclass within
CityGML.</documentation>
    </annotation>
</element>
<complexType name="GenericUnoccupiedSpaceType">
    <complexContent>
        <extension base="core:AbstractUnoccupiedSpaceType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the
GenericUnoccupiedSpace.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the
GenericUnoccupiedSpace.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">

```

```

<annotation>
    <documentation>Specifies the actual uses of the
GenericUnoccupiedSpace.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfGenericUnoccupiedSpace" type="gen:ADEOfGenericUnoccupiedSpacePropertyType">
    <annotation>
        <documentation>Augments the GenericUnoccupiedSpace with properties
defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</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="IntAttribute.PropertyType">
    <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>
<complexType name="StringAttributePropertyType">
  <sequence>
    <element ref="gen:StringAttribute"/>
  </sequence>
</complexType>
<element name="UriAttribute" substitutionGroup="core:AbstractGenericAttribute"
type="gen:UriAttributeType">
  <annotation>
    <documentation>UriAttribute is a data type used to define generic attributes of
type "URI".</documentation>
  </annotation>
</element>
<complexType name="UriAttributeType">
  <complexContent>
    <extension base="core:AbstractGenericAttributeType">
      <sequence>
        <element name="name" type="string">
          <annotation>
            <documentation>Specifies the name of the UriAttribute.</documentation>
          </annotation>
        </element>
        <element name="value" type="anyURI">
          <annotation>
            <documentation>Specifies the "URI" value.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="UriAttributePropertyType">
  <sequence>
    <element ref="gen:UriAttribute"/>
  </sequence>
</complexType>
</schema>

```

C.10. Land Use

The CityGML Land Use module is defined in the XML Schema Definition file *landUse.xsd* ([Listing](#)

38). The target namespace <http://www.opengis.net/citygml/landuse/3.0> is associated with this module.

Listing 38. landUse.xsd

```
<?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:luse="http://www.opengis.net/citygml/landuse/3.0" elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/landuse/3.0" version="3.0.0">
  <annotation>
    <documentation>The LandUse module supports representation of areas of the earth's
    surface dedicated to a specific land use.</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="ADEOfLandUse" substitutionGroup="gml:AbstractObject"
  type="luse:ADEOfLandUseType">
    <annotation>
      <documentation>ADEOfLandUse acts as a hook to define properties within an ADE
      that are to be added to a LandUse.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfLandUseType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfLandUsePropertyType">
    <sequence>
      <element ref="luse:ADEOfLandUse"/>
    </sequence>
  </complexType>
  <element name="LandUse" substitutionGroup="core:AbstractThematicSurface"
  type="luse:LandUseType">
    <annotation>
      <documentation>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.</documentation>
    </annotation>
  </element>
  <complexType name="LandUseType">
    <complexContent>
      <extension base="core:AbstractThematicSurfaceType">
        <sequence>
          <element minOccurs="0" name="class" type="gml:CodeType">
            <annotation>
              <documentation>Indicates the specific type of the
              LandUse.</documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

```

<element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
    <annotation>
        <documentation>Specifies the intended purposes of the
LandUse.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
    <annotation>
        <documentation>Specifies the actual uses of the LandUse.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfLandUse"
type="luse:ADEOfLandUsePropertyType">
    <annotation>
        <documentation>Augments the LandUse with properties defined in an
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="LandUsePropertyType">
    <sequence minOccurs="0">
        <element ref="luse:LandUse"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>
```

C.11. Point Cloud

The CityGML Point Cloud module is defined in the XML Schema Definition file *pointCloud.xsd* ([Listing 39](#)). The target namespace <http://www.opengis.net/citygml/pointcloud/3.0> is associated with this module.

Listing 39. pointCloud.xsd

```

<?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/--&gt;
&lt;element abstract="true" name="ADEOfPointCloud"
substitutionGroup="gml:AbstractObject" type="pcl:ADEOfPointCloudType"&gt;
  &lt;annotation&gt;
    &lt;documentation&gt;ADEOfPointCloud acts as a hook to define properties within an ADE
that are to be added to a PointCloud.&lt;/documentation&gt;
  &lt;/annotation&gt;
&lt;/element&gt;
&lt;complexType abstract="true" name="ADEOfPointCloudType"&gt;
  &lt;sequence/&gt;
&lt;/complexType&gt;
&lt;complexType name="ADEOfPointCloudPropertyType"&gt;
  &lt;sequence&gt;
    &lt;element ref="pcl:ADEOfPointCloud"/&gt;
  &lt;/sequence&gt;
&lt;/complexType&gt;
&lt;element name="PointCloud" substitutionGroup="core:AbstractPointCloud"
type="pcl:PointCloudType"&gt;
  &lt;annotation&gt;
    &lt;documentation&gt;A PointCloud is an unordered collection of points that is a
sampling of the geometry of a space or space boundary.&lt;/documentation&gt;
  &lt;/annotation&gt;
&lt;/element&gt;
&lt;complexType name="PointCloudType"&gt;
  &lt;complexContent&gt;
    &lt;extension base="core:AbstractPointCloudType"&gt;
      &lt;sequence&gt;
        &lt;element minOccurs="0" name="mimeType" type="gml:CodeType"&gt;
          &lt;annotation&gt;
            &lt;documentation&gt;Specifies the MIME type of the external point cloud
file.&lt;/documentation&gt;
          &lt;/annotation&gt;
        &lt;/element&gt;
        &lt;element minOccurs="0" name="pointFile" type="anyURI"&gt;
          &lt;annotation&gt;
            &lt;documentation&gt;Specifies the URI that points to the external point cloud
file.&lt;/documentation&gt;
          &lt;/annotation&gt;
        &lt;/element&gt;
        &lt;element minOccurs="0" name="pointFileSrsName" type="string"&gt;
          &lt;annotation&gt;
            &lt;documentation&gt;Indicates the coordinate reference system used by the
external point cloud file.&lt;/documentation&gt;
          &lt;/annotation&gt;
        &lt;/element&gt;
        &lt;element minOccurs="0" name="points" type="gml:MultiPointPropertyType"&gt;
          &lt;annotation&gt;
</pre>

```

```

<documentation>Relates to the 3D MultiPoint geometry of the PointCloud
stored inline with the city model.</documentation>
</annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfPointCloud"
type="pcl:ADEOfPointCloudPropertyType">
<annotation>
<documentation>Augments the PointCloud with properties defined in an
ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="PointCloud.PropertyType">
<sequence minOccurs="0">
<element ref="pcl:PointCloud"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>
```

C.12. Relief

The CityGML Relief module is defined in the XML Schema Definition file *relief.xsd* ([Listing 40](#)). The target namespace <http://www.opengis.net/citygml/relief/3.0> is associated with this module.

Listing 40. relief.xsd

```

<?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>
</element>
<complexType abstract="true" name="ADEOfAbstractReliefComponentType">
    <sequence/>
</complexType>
<complexType name="ADEOfAbstractReliefComponentPropertyType">
    <sequence>
        <element ref="dem:ADEOfAbstractReliefComponent"/>
    </sequence>
</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 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 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 abstract="true" name="ADEOfRasterReliefType">
    <sequence/>
</complexType>

```

```

<complexType name="ADEOfRasterRelief.PropertyType">
  <sequence>
    <element ref="dem:ADEOfRasterRelief"/>
  </sequence>
</complexType>
<element abstract="true" name="ADEOfReliefFeature"
substitutionGroup="gml:AbstractObject" type="dem:ADEOfReliefFeatureType">
  <annotation>
    <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>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

</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>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfAbstractReliefComponent" type="dem:ADEOfAbstractReliefComponentPropertyType">
    <annotation>
        <documentation>Augments AbstractReliefComponent with properties defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<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>
            </sequence>
        </extension>
    </complexContent>

```

```

        </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">
        <element ref="dem:MassPointRelief"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
            </sequence>
            <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="RasterReliefPropertyType">
    <sequence minOccurs="0">
        <element ref="dem:RasterRelief"/>
    </sequence>

```

```

<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="ReliefFeature" substitutionGroup="core:AbstractSpaceBoundary"
type="dem:ReliefFeatureType">
<annotation>
<documentation>A ReliefFeature is a collection of terrain components
representing the Earth's surface, also known as the Digital Terrain
Model.</documentation>
</annotation>
</element>
<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>
<documentation>A TINRelief represents a terrain component as a triangulated
irregular network.</documentation>
</annotation>
</element>
<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>
<complexType name="TINReliefPropertyType">
<sequence minOccurs="0">
<element ref="dem:TINRelief"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>

```

C.13. Transportation

The CityGML Transportation module is defined in the XML Schema Definition file *transportation.xsd* ([Listing 41](#)). The target namespace <http://www.opengis.net/citygml/transportation/3.0> is associated with this module.

Listing 41. transportation.xsd

```
<?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:tran="http://www.opengis.net/citygml/transportation/3.0"
  elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/transportation/3.0" version="3.0.0">
  <annotation>
    <documentation>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.</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="ADEOfAbstractTransportationSpace"
  substitutionGroup="gml:AbstractObject"
  type="tran:ADEOfAbstractTransportationSpaceType">
    <annotation>
      <documentation>ADEOfAbstractTransportationSpace acts as a hook to define
      properties within an ADE that are to be added to
      AbstractTransportationSpace.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAbstractTransportationSpaceType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractTransportationSpacePropertyType">
    <sequence>
      <element ref="tran:ADEOfAbstractTransportationSpace"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfAuxiliaryTrafficArea"
  substitutionGroup="gml:AbstractObject" type="tran:ADEOfAuxiliaryTrafficAreaType">
    <annotation>
      <documentation>ADEOfAuxiliaryTrafficArea acts as a hook to define properties
      within an ADE that are to be added to an AuxiliaryTrafficArea.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAuxiliaryTrafficAreaType">
    <sequence/>
```

```

    </complexType>
<complexType name="ADEOfAuxiliaryTrafficAreaPropertyType">
    <sequence>
        <element ref="tran:ADEOfAuxiliaryTrafficArea"/>
    </sequence>
</complexType>
<element abstract="true" name="ADEOfAuxiliaryTrafficSpace"
substitutionGroup="gml:AbstractObject" type="tran:ADEOfAuxiliaryTrafficSpaceType">
    <annotation>
        <documentation>ADEOfAuxiliaryTrafficSpace acts as a hook to define properties
within an ADE that are to be added to an AuxiliaryTrafficSpace.</documentation>
    </annotation>
</element>
<complexType abstract="true" name="ADEOfAuxiliaryTrafficSpaceType">
    <sequence/>
</complexType>
<complexType name="ADEOfAuxiliaryTrafficSpace.PropertyType">
    <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="ADEOfClearanceSpace.PropertyType">
    <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="ADEOfHole.PropertyType">
    <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>
<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">
<sequence>
<element ref="tran:ADEOfTrack"/>
</sequence>
</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">
<sequence>
<element ref="tran:ADEOfTrafficArea"/>
</sequence>
</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">
<sequence>
<element ref="tran:ADEOfTrafficSpace"/>
</sequence>
</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>
    <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:Occupancy.PropertyType">
          <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>
        </element>
        <complexType>
          <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
              <sequence minOccurs="0">
                <element ref="tran:TrafficSpace"/>
              </sequence>
              <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
          </complexContent>
        </complexType>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        </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>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tran:AuxiliaryTrafficSpace"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="hole">
    <annotation>
        <documentation>Relates to the holes that are part of the transportation space.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tran:Hole"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="marking">
    <annotation>
        <documentation>Relates to the markings that are part of the transportation space.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tran:Marking"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>

```

```

</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfAbstractTransportationSpace"
type="tran:ADEOfAbstractTransportationSpacePropertyType">
    <annotation>
        <documentation>Augments AbstractTransportationSpace with properties
defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

<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>Augments the AuxiliaryTrafficArea with properties defined
in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="AuxiliaryTrafficAreaPropertyType">
    <sequence minOccurs="0">
        <element ref="tran:AuxiliaryTrafficArea"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="AuxiliaryTrafficSpace"
substitutionGroup="core:AbstractUnoccupiedSpace"
type="tran:AuxiliaryTrafficSpaceType">
    <annotation>
        <documentation>An AuxiliaryTrafficSpace is a space within the transportation
space not intended for traffic purposes.</documentation>
    </annotation>
</element>
<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
and semantic decomposition.</documentation>
    </annotation>
</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>
</sequence>
</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="Hole.PropertyType">
    <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">
            <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>
        </extension>
    </complexContent>
</complexType>
<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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

</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="Marking.PropertyType">
    <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">
                    <annotation>

```

```

<documentation>Relates to the sections that are part of the
Railway.</documentation>
</annotation>
<complexType>
    <complexContent>
        <extension base="gml:AbstractFeatureMemberType">
            <sequence minOccurs="0">
                <element ref="tran:Section"/>
            </sequence>
            <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </extension>
    </complexContent>
</complexType>
</element>
<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"
type="tran:ADEOfRailwayPropertyType">
    <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">
          <annotation>
            <documentation>Indicates the specific type of the Road.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
          <annotation>
            <documentation>Specifies the intended purposes of the
Road.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
          <annotation>
            <documentation>Specifies the actual uses of the Road.</documentation>
          </annotation>
        </element>
        <element maxOccurs="unbounded" minOccurs="0" name="section">
          <annotation>
            <documentation>Relates to the sections that are part of the
Road.</documentation>
          </annotation>
        </element>
        <complexType>
          <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
              <sequence minOccurs="0">
                <element ref="tran:Section"/>
              </sequence>
              <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
          </complexContent>
        </complexType>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element maxOccurs="unbounded" minOccurs="0" name="intersection">
  <annotation>
    <documentation>Relates to the intersections that are part of the
Road.</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="adeOfRoad"
type="tran:ADEOfRoadPropertyType">
<annotation>
    <documentation>Augments the Road with properties defined in an
ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="RoadPropertyType">
<sequence minOccurs="0">
    <element ref="tran:Road"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Section" substitutionGroup="tran:AbstractTransportationSpace"
type="tran:SectionType">
<annotation>
    <documentation>A Section is a transportation space that is a segment of a Road,
Railway, Track, or Waterway.</documentation>
</annotation>
</element>
<complexType name="SectionType">
<complexContent>
    <extension base="tran:AbstractTransportationSpaceType">
        <sequence>
            <element minOccurs="0" name="class" type="gml:CodeType">
                <annotation>
                    <documentation>Indicates the specific type of the
Section.</documentation>
                </annotation>
            </element>
            <element maxOccurs="unbounded" minOccurs="0" name="adeOfSection"
type="tran:ADEOfSectionPropertyType">
                <annotation>
                    <documentation>Augments the Section with properties defined in an
ADE.</documentation>
                </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

    </complexContent>
</complexType>
<complexType name="SectionPropertyType">
    <sequence minOccurs="0">
        <element ref="tran:Section"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<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="Square.PropertyType">
  <sequence minOccurs="0">
    <element ref="tran:Square"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Track" substitutionGroup="tran:AbstractTransportationSpace"
type="tran:TrackType">
  <annotation>
    <documentation>A Track is a small path mainly used by pedestrians. Tracks can be
segmented into Sections and Intersections.</documentation>
  </annotation>
</element>
<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>
      </sequence>
      <complexType>
        <complexContent>
          <extension base="gml:AbstractFeatureMemberType">
            <sequence minOccurs="0">
              <element ref="tran:Section"/>
            </sequence>
            <attributeGroup ref="gml:AssociationAttributeGroup"/>
          </extension>
        </complexContent>
      </complexType>
    </extension>
  </complexContent>
</complexType>

```

```

</element>
<element maxOccurs="unbounded" minOccurs="0" name="intersection">
    <annotation>
        <documentation>Relates to the intersections that are part of the
Track.</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="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>
            </sequence>
        </extension>
    </complexContent>

```

```

</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"
type="gml:CodeType">
    <annotation>
        <documentation>Specifies the actual uses of the
TrafficArea.</documentation>
    </annotation>
</element>
<element minOccurs="0" name="surfaceMaterial" type="gml:CodeType">
    <annotation>
        <documentation>Specifies the type of pavement of the
TrafficArea.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfTrafficArea"
type="tran:ADEOfTrafficAreaPropertyType">
    <annotation>
        <documentation>Augments the TrafficArea with properties defined in an
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TrafficAreaPropertyType">
    <sequence minOccurs="0">
        <element ref="tran:TrafficArea"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<simpleType 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>
    </restriction>
</simpleType>

```

```

<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 directions.</documentation>
    </annotation>
</enumeration>
</restriction>
</simpleType>
<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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </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>
    <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>
            <appinfo>
                <targetElement
xmlns="http://www.opengis.net/gml/3.2">tran:TrafficSpace</targetElement>
            </appinfo>
        </annotation>
    </element>
    <element maxOccurs="unbounded" minOccurs="0" name="successor"
type="gml:ReferenceType">
        <annotation>
            <documentation>Indicates the successor(s) of the
TrafficSpace.</documentation>
            <appinfo>
                <targetElement
xmlns="http://www.opengis.net/gml/3.2">tran:TrafficSpace</targetElement>
            </appinfo>
        </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>

```

```

        </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>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TrafficSpacePropertyType">
    <sequence minOccurs="0">
        <element ref="tran:TrafficSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<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>
            </sequence>
        </extension>
    </complexContent>

```

```

</element>
<element maxOccurs="unbounded" minOccurs="0" name="section">
    <annotation>
        <documentation>Relates to the sections that are part of the Waterway.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence minOccurs="0">
                    <element ref="tran:Section"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="intersection">
    <annotation>
        <documentation>Relates to the intersections that are part of the Waterway.</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="adeOfWaterway" type="tran:ADEOfWaterwayPropertyType">
    <annotation>
        <documentation>Augments the Waterway with properties defined in an ADE.</documentation>
    </annotation>
    <complexType>
        <sequence>
            <element ref="tran:Waterway"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
</element>

```

```
</schema>
```

C.14. Tunnel

The CityGML Tunnel module is defined in the XML Schema Definition file *tunnel.xsd* ([Listing 42](#)). The target namespace <http://www.opengis.net/citygml/tunnel/3.0> is associated with this module.

Listing 42. tunnel.xsd

```
<?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"
  xmlns:tun="http://www.opengis.net/citygml/tunnel/3.0" elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/tunnel/3.0" version="3.0.0">
  <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>
<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>

```

```

</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>
<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">
    <annotation>
        <documentation>Relates the furniture objects to the Tunnel or
TunnelPart.</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="adeOfAbstractTunnel"
type="tun:ADEOfAbstractTunnelPropertyType">
    <annotation>
        <documentation>Augments AbstractTunnel with properties defined in an
ADE.</documentation>
    </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<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>
        <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>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        </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
ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="HollowSpace.PropertyType">
    <sequence minOccurs="0">
        <element ref="tun:HollowSpace"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="Tunnel" substitutionGroup="tun:AbstractTunnel" type="tun:TunnelType">
    <annotation>
        <documentation>A Tunnel represents a horizontal or sloping enclosed passage way
of a certain length, mainly underground or underwater. [cf. ISO 6707-
1]</documentation>
    </annotation>
</element>
<complexType name="TunnelType">
    <complexContent>
        <extension base="tun:AbstractTunnelType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="tunnelPart">
                    <annotation>
                        <documentation>Relates the tunnel parts to the Tunnel.</documentation>
                    </annotation>
<complexType>
                <complexContent>
                    <extension base="gml:AbstractFeatureMemberType">
                        <sequence minOccurs="0">
                            <element ref="tun:TunnelPart"/>
                        </sequence>
                        <attributeGroup ref="gml:AssociationAttributeGroup"/>
                    </extension>
                </complexContent>
</complexType>
            </sequence>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfTunnel"

```

```

type="tun:ADEOfTunnelPropertyType">
    <annotation>
        <documentation>Augments the Tunnel with properties defined in an ADE.</documentation>
    </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<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>
                <element maxOccurs="unbounded" minOccurs="0"

```

```

name="adeOfTunnelConstructiveElement"
type="tun:ADEOfTunnelConstructiveElementPropertyType">
    <annotation>
        <documentation>Augments the TunnelConstructiveElement with properties
defined in an ADE.</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>
<complexType name="TunnelFurniturePropertyType">
    <sequence minOccurs="0">
        <element ref="tun:TunnelFurniture"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TunnelInstallation" substitutionGroup="con:AbstractInstallation"
type="tun:TunnelInstallationType">
    <annotation>
        <documentation>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.</documentation>
    </annotation>
</element>
<complexType name="TunnelInstallationType">
    <complexContent>
        <extension base="con:AbstractInstallationType">
            <sequence>
                <element minOccurs="0" name="class" type="gml:CodeType">
                    <annotation>
                        <documentation>Indicates the specific type of the
TunnelInstallation.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="function"
type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the intended purposes of the
TunnelInstallation.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="usage"
type="gml:CodeType">
                    <annotation>
                        <documentation>Specifies the actual uses of the
TunnelInstallation.</documentation>
                    </annotation>
                </element>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfTunnelInstallation"
type="tun:ADEOfTunnelInstallationPropertyType">

```

```

<annotation>
    <documentation>Augments the TunnelInstallation with properties defined
in an ADE.</documentation>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TunnelInstallationPropertyType">
    <sequence minOccurs="0">
        <element ref="tun:TunnelInstallation"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="TunnelPart" substitutionGroup="tun:AbstractTunnel"
type="tun:TunnelPartType">
    <annotation>
        <documentation>A TunnelPart is a physical or functional subdivision of a Tunnel.
It would be considered a Tunnel, if it were not part of a collection of other
TunnelParts.</documentation>
    </annotation>
</element>
<complexType name="TunnelPartType">
    <complexContent>
        <extension base="tun:AbstractTunnelType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfTunnelPart"
type="tun:ADEOfTunnelPartPropertyType">
                    <annotation>
                        <documentation>Augments the TunnelPart with properties defined in an
ADE.</documentation>
                    </annotation>
</element>
            </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TunnelPartPropertyType">
    <sequence minOccurs="0">
        <element ref="tun:TunnelPart"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>

```

C.15. Vegetation

The CityGML Vegetation module is defined in the XML Schema Definition file *vegetation.xsd* ([Listing 43](#)). The target namespace <http://www.opengis.net/citygml/vegetation/3.0> is associated with this module.

Listing 43. vegetation.xsd

```
<?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"
  elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/vegetation/3.0" version="3.0.0">
  <annotation>
    <documentation>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.</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="ADEOfAbstractVegetationObject"
  substitutionGroup="gml:AbstractObject" type="veg:ADEOfAbstractVegetationObjectType">
    <annotation>
      <documentation>ADEOfAbstractVegetationObject acts as a hook to define properties within an ADE that are to be added to AbstractVegetationObject.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfAbstractVegetationObjectType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfAbstractVegetationObjectPropertyType">
    <sequence>
      <element ref="veg:ADEOfAbstractVegetationObject"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfPlantCover"
  substitutionGroup="gml:AbstractObject" type="veg:ADEOfPlantCoverType">
    <annotation>
      <documentation>ADEOfPlantCover acts as a hook to define properties within an ADE that are to be added to a PlantCover.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfPlantCoverType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfPlantCoverPropertyType">
    <sequence>
```

```

<element ref="veg:ADEOfPlantCover"/>
</sequence>
</complexType>
<element abstract="true" name="ADEOfSolitaryVegetationObject"
substitutionGroup="gml:AbstractObject" type="veg:ADEOfSolitaryVegetationObjectType">
<annotation>
  <documentation>ADEOfSolitaryVegetationObject acts as a hook to define properties
within an ADE that are to be added to a SolitaryVegetationObject.</documentation>
</annotation>
</element>
<complexType abstract="true" name="ADEOfSolitaryVegetationObjectType">
  <sequence/>
</complexType>
<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"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="SolitaryVegetationObject"
substitutionGroup="veg:AbstractVegetationObject"
type="veg:SolitaryVegetationObjectType">
    <annotation>
        <documentation>A SolitaryVegetationObject represents individual vegetation objects, e.g. trees or bushes.</documentation>
    </annotation>
</element>
<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>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

</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>
<element minOccurs="0" name="crownDiameter" type="gml:LengthType">
    <annotation>
        <documentation>Specifies the diameter of the SolitaryCityObject's crown.</documentation>
    </annotation>
</element>
<element minOccurs="0" name="rootBallDiameter" type="gml:LengthType">
    <annotation>
        <documentation>Specifies the diameter of the SolitaryCityObject's root ball.</documentation>
    </annotation>
</element>
<element minOccurs="0" name="maxRootBallDepth" type="gml:LengthType">
    <annotation>
        <documentation>Specifies the vertical distance between the lowest point of the SolitaryVegetationObject's root ball and the terrain surface.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0"
name="adeOfSolitaryVegetationObject"
type="veg:ADEOfSolitaryVegetationObjectPropertyType">
    <annotation>
        <documentation>Augments the SolitaryVegetationObject with properties defined in an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="SolitaryVegetationObjectPropertyType">
    <sequence minOccurs="0">
        <element ref="veg:SolitaryVegetationObject"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>

```

```
</schema>
```

C.16. Versioning

The CityGML Versioning module is defined in the XML Schema Definition file *versioning.xsd* ([Listing 44](#)). The target namespace <http://www.opengis.net/citygml/versioning/3.0> is associated with this module.

Listing 44. versioning.xsd

```
<?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:vers="http://www.opengis.net/citygml/versioning/3.0"
  elementFormDefault="qualified"
  targetNamespace="http://www.opengis.net/citygml/versioning/3.0" version="3.0.0">
  <annotation>
    <documentation>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.</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="ADEOfVersion" substitutionGroup="gml:AbstractObject"
    type="vers:ADEOfVersionType">
    <annotation>
      <documentation>ADEOfVersion acts as a hook to define properties within an ADE
      that are to be added to a Version.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfVersionType">
    <sequence/>
  </complexType>
  <complexType name="ADEOfVersionPropertyType">
    <sequence>
      <element ref="vers:ADEOfVersion"/>
    </sequence>
  </complexType>
  <element abstract="true" name="ADEOfVersionTransition"
    substitutionGroup="gml:AbstractObject" type="vers:ADEOfVersionTransitionType">
    <annotation>
      <documentation>ADEOfVersionTransition acts as a hook to define properties within
      an ADE that are to be added to a VersionTransition.</documentation>
    </annotation>
  </element>
  <complexType abstract="true" name="ADEOfVersionTransitionType">
    <sequence/>
```

```

    </complexType>
<complexType name="ADEOfVersionTransitionPropertyType">
    <sequence>
        <element ref="vers:ADEOfVersionTransition"/>
    </sequence>
</complexType>
<element name="Transaction" substitutionGroup="gml:AbstractObject"
type="vers:TransactionType">
    <annotation>
        <documentation>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>
    </annotation>
</element>
<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>
            <appinfo>
                <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractFeatureWithLifespan</targetElement
>
                </appinfo>
            </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>
            <appinfo>
                <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractFeatureWithLifespan</targetElement
>
                </appinfo>
            </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 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>
  </restriction>
</simpleType>

```

```

        </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>
                    <appinfo>
                        <targetElement
xmlns="http://www.opengis.net/gml/3.2">core:AbstractFeatureWithLifespan</targetElement
                    >
                
```

```

        </appinfo>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="gml:AbstractFeatureMemberType">
                <sequence/>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </extension>
        </complexContent>
    </complexType>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfVersion"
type="vers:ADEOfVersion.PropertyType">
    <annotation>
        <documentation>Augments the Version with properties defined in an
ADE.</documentation>
    </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="Version.PropertyType">
    <sequence minOccurs="0">
        <element ref="vers:Version"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</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>
    </element>
<complexType name="VersionTransitionType">
    <complexContent>
        <extension base="core:AbstractVersionTransitionType">
            <sequence>
                <element minOccurs="0" name="reason" type="string">
                    <annotation>
                        <documentation>Specifies why the VersionTransition has been carried
out.</documentation>
                    </annotation>
                </element>
                <element name="clonePredecessor" type="boolean">
                    <annotation>
                        <documentation>Indicates whether the set of city object instances
belonging to the successor version of the city model is either explicitly enumerated

```

```

within the successor version object (attribute clonePredecessor=false), or has to be
derived from the modifications of the city model provided as a list of transactions on
the city object versions contained in the predecessor version (attribute
clonePredecessor=true).</documentation>
    </annotation>
</element>
<element minOccurs="0" name="type" type="vers:TransitionTypeValueType">
    <annotation>
        <documentation>Indicates the specific type of the
VersionTransition.</documentation>
    </annotation>
</element>
<element minOccurs="0" name="from" type="gml:ReferenceType">
    <annotation>
        <documentation>Relates to the predecessor version of the
VersionTransition.</documentation>
    <appinfo>
        <targetElement
xmlns="http://www.opengis.net/gml/3.2">vers:Version</targetElement>
        </appinfo>
    </annotation>
</element>
<element minOccurs="0" name="to" type="gml:ReferenceType">
    <annotation>
        <documentation>Relates to the successor version of the
VersionTransition.</documentation>
    <appinfo>
        <targetElement
xmlns="http://www.opengis.net/gml/3.2">vers:Version</targetElement>
        </appinfo>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="transaction"
type="vers:TransactionPropertyType">
    <annotation>
        <documentation>Relates to all transactions that have been applied as
part of the VersionTransition.</documentation>
    </annotation>
</element>
<element maxOccurs="unbounded" minOccurs="0" name="adeOfVersionTransition"
type="vers:ADEOfVersionTransitionPropertyType">
    <annotation>
        <documentation>Augments the VersionTransition with properties defined in
an ADE.</documentation>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="VersionTransitionPropertyType">

```

```

<sequence minOccurs="0">
    <element ref="vers:VersionTransition"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
<attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>

```

C.17. Water Body

The CityGML Water Body module is defined in the XML Schema Definition file *waterBody.xsd* ([Listing 45](#)). The target namespace <http://www.opengis.net/citygml/waterbody/3.0> is associated with this module.

Listing 45. waterBody.xsd

```

<?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>
  </element>
  <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>
</element>
<complexType abstract="true" name="ADEOfWaterBodyType">
  <sequence/>
</complexType>
<complexType name="ADEOfWaterBodyPropertyType">
  <sequence>
    <element ref="wtr:ADEOfWaterBody"/>
  </sequence>
</complexType>
<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="ADEOfWaterGroundSurface.PropertyType">
  <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="ADEOfWaterSurface.PropertyType">
  <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>
<complexType name="WaterBodyPropertyType">
    <sequence minOccurs="0">
        <element ref="wtr:WaterBody"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="WaterGroundSurface"
substitutionGroup="wtr:AbstractWaterBoundarySurface"
type="wtr:WaterGroundSurfaceType">
    <annotation>
        <documentation>A WaterGroundSurface represents the exterior boundary surface of
the submerged bottom of a water body.</documentation>
    </annotation>
    </element>
<complexType name="WaterGroundSurfaceType">
    <complexContent>
        <extension base="wtr:AbstractWaterBoundarySurfaceType">
            <sequence>
                <element maxOccurs="unbounded" minOccurs="0" name="adeOfWaterGroundSurface"
type="wtr:ADEOfWaterGroundSurfacePropertyType">
                    <annotation>
                        <documentation>Augments the WaterGroundSurface with properties defined
in an ADE.</documentation>
                    </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
<complexType name="WaterGroundSurfacePropertyType">
    <sequence minOccurs="0">
        <element ref="wtr:WaterGroundSurface"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>

```

```

    </complexType>
    <element name="WaterSurface" substitutionGroup="wtr:AbstractWaterBoundarySurface"
type="wtr:WaterSurfaceType">
        <annotation>
            <documentation>A WaterSurface represents the upper exterior interface between a
water body and the atmosphere.</documentation>
        </annotation>
    </element>
    <complexType name="WaterSurfaceType">
        <complexContent>
            <extension base="wtr:AbstractWaterBoundarySurfaceType">
                <sequence>
                    <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>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="WaterSurfacePropertyType">
        <sequence minOccurs="0">
            <element ref="wtr:WaterSurface"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
        <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
</schema>

```

Annex D: CityGML 3.0 Conceptual Model Conformance (Normative)

A requirement of any implementation of the CityGML 3.0 Conceptual Model (CM) is that the implementation shall conform to the CM. This Annex explains how this CityGML 3.0 GML Encoding insures conformance with the CM.

D.1. Conformance of this Implementation Specification to the CityGML-3.0 CM

Implementation Specifications define how a Conceptual Model should be implemented using a specific technology. This standard defines the platform-specific implementation for the GML standards version 3.2 and 3.3, which are based on XML. The GML schemas, expressed as XML Schema Description (XSD) language files, were derived fully automatically from the UML Conceptual Model using the software tool [ShapeChange](#) with the [configuration file](#) included in this standard. **Conformance of a CityGML 3.0 GML-encoded instance document to the CityGML 3.0 Conceptual Model is assured if the instance conforms to these GML schemas, as detailed in [Table 1](#).**

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 in the UML Conceptual Model (CM) and a `gml:CodeValue` datatype in this GML Encoding Specification. 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.

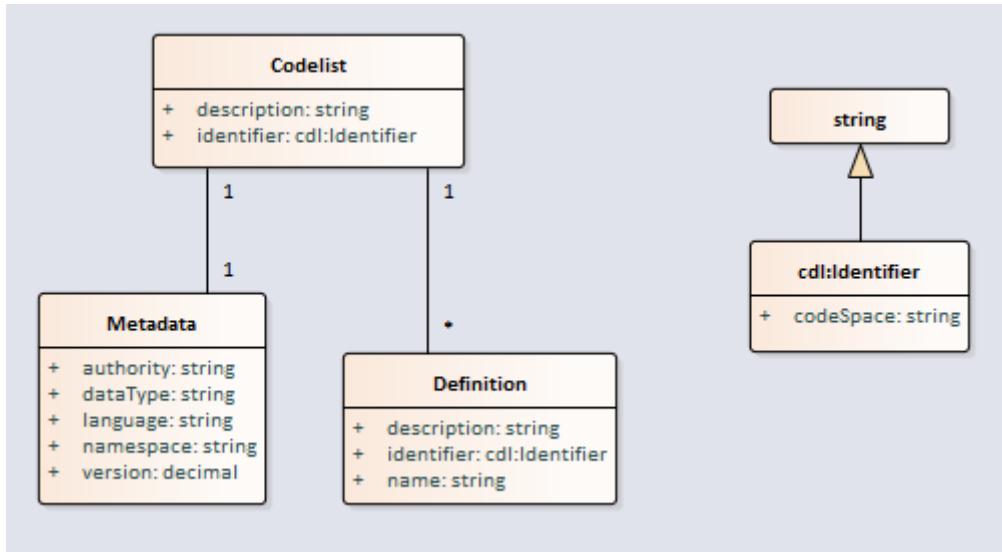


Figure 28. Suggested Codelist UML Class Diagram

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:

```

<?xml version="1.0" encoding="utf-8"?>
<gml:Dictionary gml:id="id0" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:cmd="http://www.opengis.net/citygml/codelist-metadata">
  <gml:metaDataProperty>
    <cmd:CodeListMetaData>
      <cmd:dataType>RoofTypeValue</cmd:dataType>
      <cmd:namespace>http://www.opengis.net/citygml/building/3.0</cmd:namespace>
      <cmd:language>en</cmd:language>
      <cmd:authority>Some Organization</cmd:authority>
      <cmd:version>3.0</cmd:version>
    </cmd:CodeListMetaData>
  </gml:metaDataProperty>
  <gml:description>Roof type values</gml:description>
  <gml:identifier
    codeSpace="https://ogc.org/citygml/3.0/codelists/gml/rooftypes">RoofTypeValue</gml:identifier>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="id1">
      <gml:description>roof primarily a single plane, not necessarily
      level</gml:description>
      <gml:identifier
        codeSpace="https://ogc.org/citygml/3.0/codelists/gml/rooftypes">1000</gml:identifier>
      <gml:name>flat roof</gml:name>
    </gml:Definition>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="id2">
      <gml:description>a roof that has a ridge and two gables</gml:description>
      <gml:identifier
        codeSpace="https://ogc.org/citygml/3.0/codelists/gml/rooftypes">3100</gml:identifier>
      <gml:name>saddle roof</gml:name>
    </gml:Definition>
  </gml:dictionaryEntry>
</gml:Dictionary>

```

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 '|':

CodeType	Identifier	Description
Meta CityGML_DataType	Building RoofTypeValue	
Meta CityGML_Namespace	http://www.opengis.net/citygml/building/3.0	
Meta Language	en-UK	
Meta CityGML_Version	3.0	
Code 1000	flat roof	
Code 3100	saddle roof	

E.2.3. JSON

A suggested JSON encoding of the example logical model for codelists has a JSON-Schema 2019-09 definition as follows:

```
{
  "$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"
        ]
      }
    }
  }
},
```

```
"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": [
            "@id",
            "description",
            "identifier",
            "name"
        ]
    }
},
"required": [
    "Definition"
]
},
{
    "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": [
                "@id",
                "description",
                "identifier",
                "name"
            ]
        }
    }
}
```

```
        },
        "required": [
            "Definition"
        ]
    }
}
},
"required": [
    "@id",
    "metaDataProperty",
    "description",
    "identifier",
    "dictionaryEntry"
]
}
},
"required": [
    "Dictionary"
]
}
```

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 46](#). 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 46. Use of GML 3.3 concepts in a CityGML GML instance document.

```
<CityModel xmlns:gmlce="http://www.opengis.net/gml/3.3/ce"
  ... further namespaces ...
  xsi:schemaLocation="http://www.opengis.net/gml/3.3/ce
  http://schemas.opengis.net/gml/3.3/geometryCompact.xsd
  ... further schema location entries ...">
  <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.

Recommendation 4	/rec/appearance/parameterizedtexture-gml-3.3
-------------------------	---

Assigning texture coordinates to a compact surface geometry of GML 3.3 using ParameterizedTexture elements should adhere to the following recommendations:

A	The <i>textureCoordinates</i> property SHOULD be used in the same way as for a normal surface geometry and as defined by <i>/req/appearance/parameterizedtexture</i> . The number of 2D points in the <i>textureCoordinates</i> element therefore must be identical with the number of 3D points of the simplified geometry.
---	--

B	In contrast to a <i>gml:LinearRing</i> , 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.
C	The <i>ring</i> property SHOULD be an empty element.

Annex G: Implementation Decisions (Informative)

As described below, an Implementation Model is used for deriving the GML schemas using ShapeChange. The Implementation model was created based on the [CityGML 3.0 Conceptual Model Enterprise Architect file](#).

The Enterprise Architect file of the Implementation Model and the ShapeChange configuration file required by ShapeChange for deriving the GML schemas are provided in the [folder encoding on the CityGML 3.0 GML Encoding GitHub repository](#). The Implementation Model and the ShapeChange configuration file can be used to derive the GML schemas with [ShapeChange](#) version 2.11 or later.

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](#).

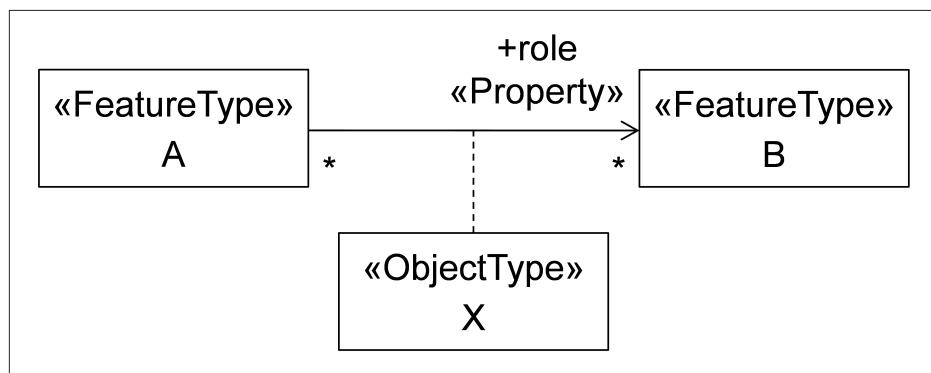


Figure 29. UML object diagram for the building in [Figure 12](#).

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.

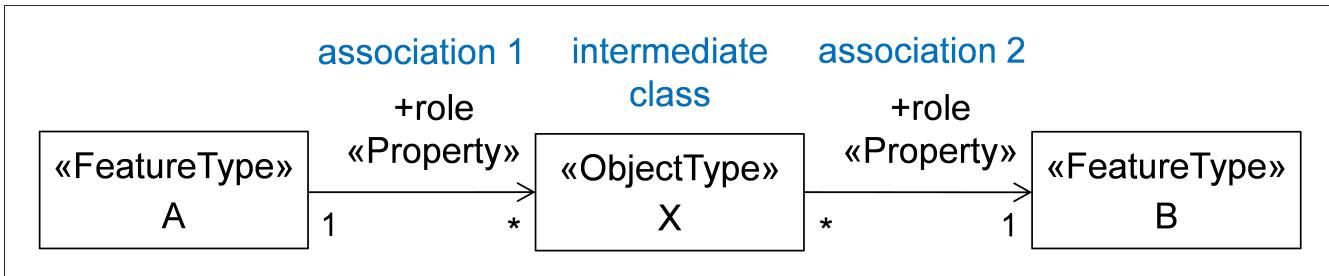


Figure 30. UML model resulting from applying the encoding rule.

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.

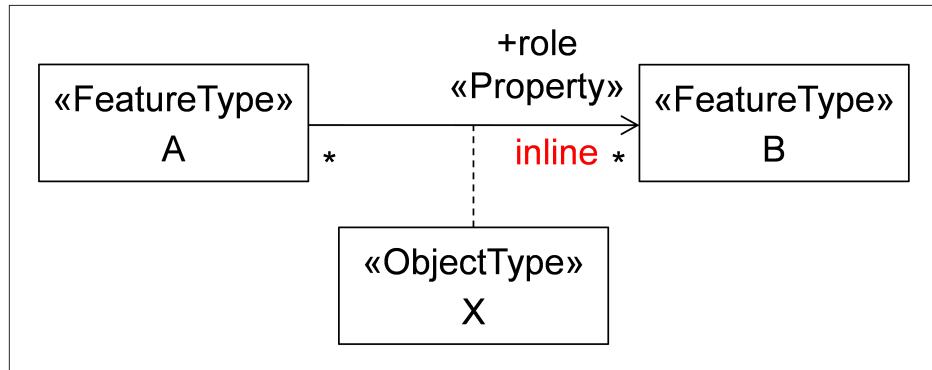


Figure 31. Association class with the tagged value “`inlineOrByReference`” set to “`inline`”.

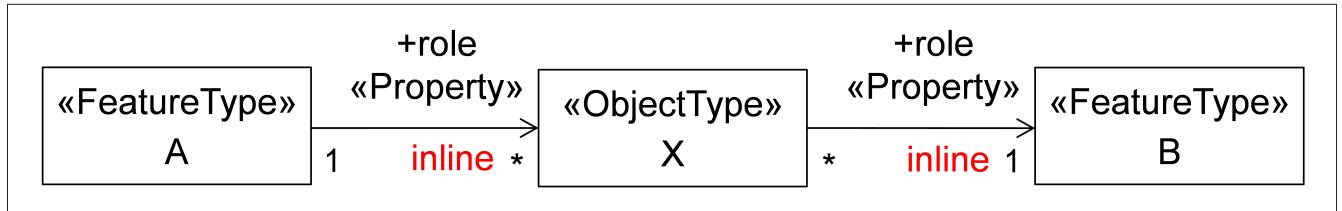


Figure 32. “`inlineOrByReference`” settings after applying the encoding rule.

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>`, ``, 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 47](#)) 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 47](#). GML instance document for the value “`inline`”.

```

<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 48](#)) results from setting the tagged value to “`byReference`”. Here, element A references element X and element X references element B using XLink.

Listing 48. GML instance document for the value “byReference”.

```
<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 49](#)) 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 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 49. GML instance document for the value “inlineOrByReference”.

```
<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 49](#). 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.)

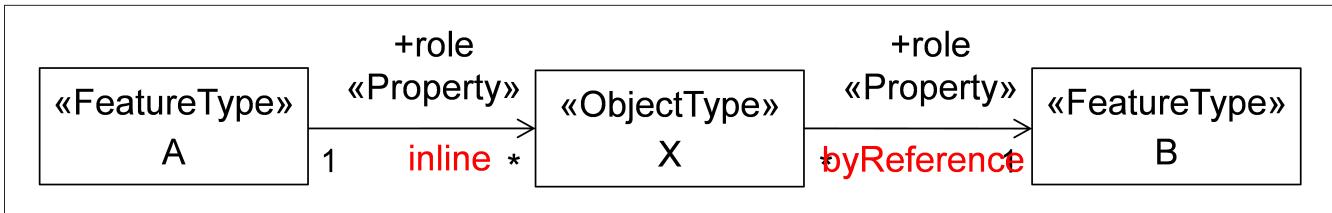


Figure 33. Desired settings for the value “`inlineOrByReference`” after applying the encoding rule.

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 49 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`”.

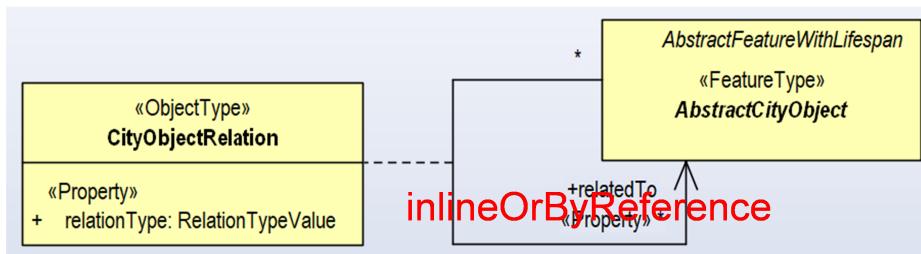


Figure 34. Association class `CityObjectRelation`.

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.

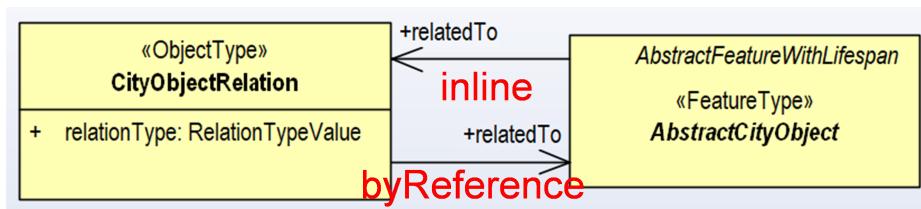


Figure 35. Association class `CityObjectRelation` represented as intermediate class.

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 50. 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 50. GML instance document for the association class CityObjectRelation.

```
<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>
      <lod2MultiSurface> ... </lod2MultiSurface>
    </con:WallSurface>
  </boundary>
</bldg:Building>
<bldg:Building gml:id="bldg_2">
  <boundary>
    <con:WallSurface gml:id="bldg_2_ws_4">
      <relatedTo>
        <CityObjectRelation>
          <relationType>equal</relationType>
          <relatedTo xlink:href="#bldg_1_ws_2"/>
        </CityObjectRelation>
      </relatedTo>
      <lod2MultiSurface> ... </lod2MultiSurface>
    </con:WallSurface>
  </boundary>
</bldg:Building>
```

G.2. Encoding of list types

Based on Recommendation 11 given in section 8.2.9 of the [OGC Testbed-17: Model-Driven Standards Engineering Report](#), the basic types *TransformationMatrix2x2*, *TransformationMatrix3x4*, *TransformationMatrix4x4*, *Color*, and *ColorPlusOpacity* were revised in the Implementation Model.

These basic types are no longer modelled as subclasses of the basic types *DoubleList* and *DoubleBetween0and1List*, respectively. Instead, they provide now a single *list* attribute themselves with a maximum multiplicity greater than 1 allowing to encode them as list-based XML Schema simple types.

The basic type *DoubleBetween0and1List* was removed from the Implementation Model as it is no longer needed. The basic type *DoubleList*, however, remains in the Implementation Model as it serves as data type for the attribute *textureCoordinates* in the class *TexCoordList* of the *Appearance* module.

By applying these recommendations in the Implementation Model, [ShapeChange](#) version 2.11 or later can be used to derive the GML schemas.

Annex H: Revision History

Date	Release	Editor	Primary clauses modified	Description
2021-12-15	0.1.0	T. Kutzner	all	initial version
2022-02-01	0.2.0	T. Kutzner, S. Smyth	Conformance, Codelist Annex	Most sections are complete, at least in draft form.
2022-07-21	0.6.0	T. Kutzner, P. Parslow, D. Vinasco-Alvarez, C. Nagel, T. Kolbe, C.S. Smyth	Conformance and general editorial.	Complete for public review
2022-07-21	0.7.0	T. Kutzner, P. Parslow, C. Nagel, C.S. Smyth	Abstract Tests, Conformance	Front material, References, general editorial.
2022-09-29	0.8.0	T. Kutzner, P. Parslow, N. Ishimaru, C.S. Smyth	Abstract Tests, Conformance	Revisions from resolution of public review comments.
2022-11-17	0.9.0	T. Kutzner, P. Parslow, C. Nagel, C.S. Smyth	Abstract Tests, Conformance	Revisions from resolution of public review comments.
2022-12-28	0.9.1	T. Kutzner, C.S. Smyth	Conformance, entire document	General editorial - from GitHub issue tasks.
2023-01-17	1.0.0	T. Kutzner, C. Nagel, C.S. Smyth	Conformance	Final cleanup prior to release by SWG.
2023-01-18	1.0.1	T. Kutzner	Annex G, Implementation Decisions, Annex D	Improve text and fix links.
2023-01-18	1.0.2	C.S. Smyth	Version and date	Correct

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

[ISO 19101-2:2018, definition 3.21] / [VIM:1993, 2.1]

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
- Agugiaro, G., Benner, J., Cipriano, P., Nouvel, R., 2018: **The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations**. Open Geospatial Data, Software and Standards, Vol. 3. <https://doi.org/10.1186/s40965-018-0042-y>
- Akahoshi, K., Ishimaru, N., Kurokawa, C., Tanaka, Y., Oishi, T., Kutzner, T., Kolbe, T. H., 2020: **I-Urban Revitalization: Conceptual modeling, implementation, and visualization towards sustainable urban planning using CityGML**. In: Proceedings of the 24th ISPRS Congress, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. V-4-2020, ISPRS. <https://doi.org/10.5194/isprs-annals-V-4-2020-179-2020>
- Becker, T., Nagel, C., Kolbe, T. H., 2011: **Integrated 3D Modeling of Multi-utility Networks and their Interdependencies for Critical Infrastructure Analysis**. In: T. H. Kolbe, G. König, C. Nagel (Eds.): Advances in 3D Geoinformation Sciences. LNG&C, Springer, Berlin. https://doi.org/10.1007/978-3-642-12670-3_1
- Beil, C., Kolbe, T. H., 2017: **CityGML and the streets of New York - A proposal for detailed street space modelling**. In: Proceedings of the 12th International 3D GeoInfo Conference 2017, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W5, ISPRS. <http://doi.org/10.5194/isprs-annals-IV-4-W5-9-2017>
- Beil, C., Kutzner, T., Schwab, B., Willenborg, B., Gawronski, A., Kolbe, T. H., 2021: **Integration of 3D point clouds with semantic 3D city models – Providing semantic information beyond classification**. In: Proceedings of the 16th International 3D GeoInfo Conference 2021, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. VIII-4/W2-2021, ISPRS. <https://doi.org/10.5194/isprs-annals-VIII-4-W2-2021-105-2021>
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015: **Applications of 3D City Models: State of the Art Review**. ISPRS International Journal of Geo-Information, 4(4). <https://doi.org/10.3390/ijgi4042842>
- Biljecki, F., Kumar, K., Nagel, C., 2018: **CityGML Application Domain Extension (ADE): overview of developments**. Open Geospatial Data, Software and Standards, 3(1). <https://doi.org/10.1186/s40965-018-0055-6>
- Billen, R., Zaki, C. E., Servières, M., Moreau, G., Hallot, P., 2012: **Developing an ontology of space: Application to 3D city modeling**. In: Leduc, T., Moreau, G., Billen, R. (eds): Usage, usability, and utility of 3D city models — European COST Action TU0801, EDP Sciences, Nantes, Vol. 02007. <https://hal.archives-ouvertes.fr/hal-01521445>
- Chaturvedi, K., Smyth, C. S., Gesquièrre, G., Kutzner, T., Kolbe, T. H., 2015: **Managing versions and history within semantic 3D city models for the next generation of CityGML**. In: Selected papers from the 10th International 3DGeoInfo Conference 2015 in Kuala Lumpur, Malaysia, Springer LNG&C, Berlin. https://doi.org/10.1007/978-3-319-25691-7_11
- Chaturvedi, K., Kolbe, T. H., 2016: **Integrating Dynamic Data and Sensors with Semantic 3D City Models in the context of Smart Cities**. In: Proceedings of the 11th International 3D Geoinfo Conference, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-2/W1, ISPRS. <https://doi.org/10.5194/isprs-annals-IV-2-W1-31-2016>

- Chaturvedi, K., Kolbe, T. H., 2017: **Future City Pilot 1 Engineering Report**, Open Geospatial Consortium. [OGC Doc. 19-098](#)
- Chaturvedi, K., Kolbe, T. H., 2019: **A Requirement Analysis on Extending Semantic 3D City Models for Supporting Time-dependent Properties**. In: Proceedings of the 4th International Conference on Smart Data and Smart Cities, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W9, ISPRS. <https://doi.org/10.5194/isprs-annals-IV-4-W9-19-2019>
- Chaturvedi, K., 2021: **Integration and Management of Time-dependent Properties with Semantic 3D City Models**. Dissertation, Technical University of Munich. <https://mediatum.ub.tum.de/node?id=1542959>
- Elfes, A., 1989: **Using occupancy grids for mobile robot perception and navigation**. Computer 22(6):46–57. <https://doi.org/10.1109/2.30720>
- Foley, J., van Dam, A., Feiner, S., Hughes, J., 2002: **Computer Graphics: Principles and Practice**. 2nd ed., Addison Wesley
- Gröger, G., Plümer, L., 2012: **CityGML – Interoperable semantic 3D city models**. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 71, July 2012. <https://dx.doi.org/10.1016/j.isprsjprs.2012.04.004>
- Gröger, G., Kolbe, T. H., Nagel, C., Häfele, K.-H., 2012: **OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0**, Open Geospatial Consortium. [OGC Doc. 12-019](#)
- Ishimaru, N., Kurokawa, C., Tanaka, Y., Oishi, T., Akahoshi, K., Kutzner, T., Kolbe, T. H., 2020: **CityGML Urban Planning ADE for i-Urban Revitalization**, Open Geospatial Consortium. [OGC Doc. 20-000r1](#)
- INSPIRE: D2.8.III.2 Data Specification on Buildings – Technical Guidelines. European Commission Joint Research Centre.
- ISO: ISO 19103:2015, Geographic Information – Conceptual Schema Language
- ISO: ISO 19107:2003, Geographic Information – Spatial Schema
- ISO: ISO/IEC 19505-2:2012, Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure
- ISO/IEC 19507:2012, Information technology — Object Management Group Object Constraint Language (OCL)
- Jensen, Christian S. and Dyreson, Curtis E.: **The Consensus Glossary of Temporal Database Concepts**. February 1998 Version. In: Temporal Databases: Research and Practice [online]. Springer Berlin Heidelberg, 1998. p. 367–405. Lecture Notes in Computer Science. Available from: 10.1007/BFb0053710
- Jensen, Christian S. and Snodgrass, Richard T., eds.: **TR-90, Temporal Database Entries for the Springer Encyclopedia of Database Systems**. Technical Report. TimeCenter, 22 May 2008. Available from: <http://timecenter.cs.aau.dk/TimeCenterPublications/TR-90.pdf>
- Johnson, Tom: **Bitemporal Data**. Elsevier, 2014. ISBN 978-0-12-408067-6. Available from: 10.1016/C2012-0-06609-4
- Kaden, R., Clemen, C., 2017: **Applying Geodetic Coordinate Reference Systems within Building Information Modeling (BIM)**. In: Proceedings of the FIG Working Week 2017,

https://www.fig.net/resources/proceedings/fig_proceedings/fig2017/papers/ts06h_TS06H_kaden_clemen_8967.pdf

- Kolbe, T. H., Gröger, G., 2003: **Towards unified 3D city models.** In: Proceedings of the Joint ISPRS Commission IV Workshop on Challenges in Geospatial Analysis, Integration and Visualization II, Stuttgart, Germany. <https://mediatum.ub.tum.de/doc/1145769/>
- Kolbe, T. H., 2009: **Representing and Exchanging 3D City Models with CityGML.** In: J. Lee, S. Zlatanova (Eds.), 3D Geo-Information Sciences, Selected Papers of the 3rd International Workshop on 3D Geo-Information in Seoul, Korea. Springer, Berlin. https://doi.org/10.1007/978-3-540-87395-2_2
- Konde, A., Tauscher, H., Biljecki, F., Crawford, J., 2018: **Floor plans in CityGML.** In: Proceedings of the 13th 3D GeoInfo Conference 2018, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W6, 25–32, ISPRS. <https://doi.org/10.5194/isprs-annals-IV-4-W6-25-2018>
- Kutzner, T., Hijazi, I., Kolbe, T. H., 2018: **Semantic Modelling of 3D Multi-utility Networks for Urban Analyses and Simulations – The CityGML Utility Network ADE.** International Journal of 3-D Information Modeling (IJ3DIM) 7(2), 1-34. <https://dx.doi.org/10.4018/IJ3DIM.2018040101>
- Kutzner, T., Chaturvedi, K. & Kolbe, T. H., 2020: **CityGML 3.0: New Functions Open Up New Applications.** PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 88, 43–61. <https://doi.org/10.1007/s41064-020-00095-z>
- Labetski, A., van Gerwen, S., Tamminga, G., Ledoux, H., Stoter, J., 2018: **A proposal for an improved transportation model in CityGML.** In: Proceedings of the 13th 3D GeoInfo Conference 2018, ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLII-4/W10, 89–96. <https://doi.org/10.5194/isprs-archives-XLII-4-W10-89-2018>
- Liu, Ling and Özsü, M. Tamer, eds.: **Encyclopedia of Database Systems.** New York, NY : Springer New York, 2018. ISBN 978-1-4614-8266-6. Available from: 10.1007/978-1-4614-8265-9
- Löwner, M.-O., Gröger, G., Benner, J., Biljecki, F., Nagel, C., 2016: **Proposal for a new LOD and multi-representation concept for CityGML.** In: Proceedings of the 11th 3D Geoinfo Conference 2016, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-2/W1, 3–12. <https://doi.org/10.5194/isprs-annals-IV-2-W1-3-2016>
- Nouvel, R., Bahu, J. M., Kaden, R., Kaempf, J., Cipriano, P., Lauster, M., Haefele, K.-H., Munoz, E., Tournaire, O., Casper, E., 2015: **Development of the CityGML Application Domain Extension Energy for Urban Energy Simulation.** In: Proceedings of Building Simulation 2015 - 14th Conference of the International Building Performance Simulation Association, IBPSA, 559-564. <http://www.ibpsa.org/proceedings/BS2015/p2863.pdf>
- OGC: OGC 10-129r1, OGC® Geography Markup Language (GML) — Extended schemas and encoding rules, v3.3, 2012.
- Smith, B., Varzi, A. C., 2000: **Fiat and Bona Fide Boundaries.** Philosophy and Phenomenological Research, Vol. 60, No. 2, 401-420. <https://doi.org/10.2307/2653492>
- Snodgrass, Richard T: **Developing time-oriented database applications in SQL.** San Francisco, California : Morgan Kaufmann Publishers, July 1999. ISBN 1-55860-436-7. Available from: <http://www.cs.arizona.edu/rts/dbbook.pdf>

- Stadler, A., Kolbe, T. H., 2007: **Spatio-semantic Coherence in the Integration of 3D City Models**. In: Proceedings of the 5th International ISPRS Symposium on Spatial Data Quality ISSDQ 2007 in Enschede. http://www.isprs.org/proceedings/XXXVI/2-C43/Session1/paper_Stadler.pdf
- Vretanos, P. A. 2010: **OpenGIS Web Feature Service 2.0 Interface Standard**, Open Geospatial Consortium. [OGC Doc. 09-025r1](#)
- OASIS MQTT Technical Committee: **MQTT Version 5.0 Standard**, OASIS, March 7, 2019, Available from [OASIS](#).
- Reed, C., Belayneh T.: **OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package Format Specification**, Open Geospatial Consortium, Available from [OGC Doc. 17-014r7](#)
- [[3dtiles_citation, OGC 3D Tiles]]Cozzi, P., Lilley, S., Getz, G. **OGC 3D Tiles Specification 1.0** Open Geospatial Consortium, Available from [OGC Doc. 18-053r2](#)
- Burggraf, D.: **OGC KML 2.3**, Open Geospatial Consortium, Available from [OGC Doc. 12-007r2](#)
- Bröring, A., Stasch, C., Echterhoff, J.: **OGC® Sensor Observation Service Interface Standard**, Open Geospatial Consortium, Available from [OGC Doc. 12-006](#)
- Liang, S., Huang, C., Khalafbeigi, T.: **OGC SensorThings API Part 1: Sensing**, Open Geospatial Consortium, Available from [OGC Doc. 15-078r6](#)
- [[3dps_citation, OGC 3D Portrayal Service]]Hagedorn, B., Thum, S., Reitz, T., Coors, V., Gutbell, R.: **OGC® 3D Portrayal Service 1.0**, Open Geospatial Consortium, Available from [OGC Doc. 15-001r3](#).
- Bhatia, S., Cozzi, P., Knyazev, A., Parisi, T.: **The GL Transmission Format (glTF)**, The Khronos Group, Available from <https://www.khronos.org/gltf>.