Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2020-08-05>

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

Internal reference number of this OGC® document: 20-066

Version: 0.1

Category: OGC® Engineering Guidance

Editor: Charles Heazel

OGC City Geography Markup Language (CityGML) 3.0 Conceptual Model Users Guide

Warning

This document is not an OGC Standard. This document provides Engineering Guidance on the use of the PGC CityGML: 3.0 Conceptual Model Standard. This document is a non-normative resource and not an official position of the OGC membership. It is subject to change without notice and may not be referred to as an OGC Standard. Further, Engineering Guidance should not be referenced as required or mandatory technology in procurements.

Document type:	OGC®Engineering Guidance
Document subtype:	
Document stage:	Draft
Document language:	English

License Agreement

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

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

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

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

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

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

Table of Contents

i. Abstract

CityGML is an open conceptual data model for the storage and exchange of virtual 3D city models. It is defined through a Unified Modeling Language (UML) object model. This UML model extends the ISO Technical Committee 211 (TC211) conceptual model standards for spatial and temporal data. Building on the ISO fundation assures that the man-made features described in the City Models share the same spatial-temporal universe as the surrounding countryside within which they reside.

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

ii. Keywords

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

ogcdoc, OGC document, CityGML, 3D city models

iii. Preface

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

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

Chapter 1. Introduction

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

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

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

CityGML is an open conceptual data model for the storage and exchange of virtual 3D city models. It is defined through a Unified Modeling Language (UML) object model. This UML model extends the ISO Technical Committee 211 (TC211) conceptual model standards for spatial and temporal data. Building on the ISO fundation assures that the man-made features described in the City Models share the same spatial-temporal universe as the surrounding countryside within which they reside.

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

Chapter 2. Scope

This document provides Engineering Guidance on the use of the CityGML 3.0 Conceptual Model Standard.

The OGC Conceptual Model Standard specifies the representation of virtual 3D city and landscape models. The CityGML 3.0 Conceptual Model is expected to be the basis for a number of future Encoding Standards in which subsets of the Conceptual Model can be implemented. These Encoding Standards will enable both storage and exchange of data.

The CityGML 3.0 Conceptual Model Standard was designed to be concise and easy to use. As a result, most non-normative content has been removed. The purpose of this Users Guide is to capture that non-normative content and make it easy to access if and when needed.

Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of OGC 20-010. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of OGC 12-019 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

- IETF: RFC 2045 & 2046, Multipurpose Internet Mail Extensions (MIME). (November 1996),
- IETF: RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. (January 2005)
- ISO: ISO 8601:2004, Data elements and interchange formats – Information interchange – Representation of dates and times
- ISO: ISO 19103:2015, Geographic Information – Conceptual Schema Language
- ISO: ISO 19107:2003, Geographic Information – Spatial Schema
- ISO: ISO 19111:2019, Geographic information – Referencing by coordinates
- ISO: ISO/IEC 19505-2:2012, Information technology — Object Management Group Unified ModelingLanguage (OMG UML) — Part 2: Superstructure
- ISO/IEC 19507:2012, Information technology — Object Management Group Object Constraint Language (OCL)
- 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

Terms and Definitions

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

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

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]

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

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

conformance test class
set of conformance test modules that must be applied to receive a single certificate of conformance
[OGC 08-131r3, definition 4.4]

feature
abstraction of real world phenomena
[ISO 19101-1:2014, definition 4.1.11]

feature attribute
characteristic of a feature
[ISO 19101-1:2014, definition 4.1.12]

feature type
class of features having common characteristics
[ISO 19156:2011, definition 4.7]

level 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]

measurement
set of operations having the object of determining the value of a quantity
[ISO 19101-2:2018, definition 3.21] / [VIM:1993, 2.1]

model
abstraction of some aspects of reality
[ISO 19109:2015, definition 4.15]

observation
act of measuring or otherwise determining the value of a property
[ISO 19156:2011, definition 4.11]

observation procedure
method, algorithm or instrument, or system of these, which may be used in making an observation
[ISO 19156:2011, 4.12]

observation result
estimate of the value of a property determined through a known observation procedure
[ISO 19156:2011, 4.14]

property
facet or attribute of an object referenced by a name.
[ISO 19143:2010, definition 4.21]

requirements class
aggregate of all requirement modules that must all be satisfied to satisfy a conformance test class
[OGC 08-131r3, definition 4.19]

schema
formal description of a model
[ISO 19101-1:2014, definition 4.1.34]

sensor
type of observation procedure that provides the estimated value of an observed property at its output
[OGC 08-094r1, definition 4.5]

timeseries
sequence of data values which are ordered in time
[OGC 15-043r3]

universe of discourse
view of the real or hypothetical world that includes everything of interest
[ISO 19101-1:2014, definition 4.1.38]

version
Particular variation of a spatial object
[INSPIRE Glossary]

Abbreviated Terms

The following abbreviated terms are used in this document:

The list of acronyms needs to be reviewed once all sections have been updated.

- 2D Two Dimensional
- 3D Three Dimensional
- AEC Architecture, Engineering, Construction
- ALKIS German National Standard for Cadastral Information
- ATKIS German National Standard for Topographic and Cartographic Information
- B-Rep Boundary Representation
- bSI buildingSMART International
- CAD Computer Aided Design
- COLLADA Collaborative Design Activity
- CSG Constructive Solid Geometry
- DTM Digital Terrain Model
- DXF Drawing Exchange Format
- EuroSDR European Spatial Data Research Organisation
- ESRI Environmental Systems Research Institute
- FM Facility Management
- GDF Geographic Data Files
- GDI-DE Spatial Data Infrastructure Germany (Geodateninfrastruktur Deutschland)
- GDI NRW Geodata Infrastructure North-Rhine Westphalia
- GML Geography Markup Language
- IAI International Alliance for Interoperability (now buildingSMART International (bSI))
- IETF Internet Engineering Task Force
- IFC Industry Foundation Classes
- ISO International Organization for Standardisation
- LOD Level of Detail
- NBIMS National Building Information Model Standard

- OASIS Organisation for the Advancement of Structured Information Standards
- OGC Open Geospatial Consortium
- OSCRE Open Standards Consortium for Real Estate
- SIG 3D Special Interest Group 3D of the GDI-DE
- TC211 ISO Technical Committee 211
- TIC Terrain Intersection Curve
- TIN Triangulated Irregular Network
- UML Unified Modeling Language
- URI Uniform Resource Identifier
- VRML Virtual Reality Modeling Language
- W3C World Wide Web Consortium
- W3DS OGC Web 3D Service
- WFS OGC Web Feature Service
- X3D Open Standards XML-enabled 3D file format of the Web 3D Consortium
- XML Extensible Markup Language
- xAL OASIS extensible Address Language

Chapter 4. User Guide

4.1. Conceptual Modeling

NOTE	Under Construction

4.2. Spatial-Temporal Fundamentals

NOTE	Under Construction

4.3. Extensions

Chapter 5. CityGML Extension Mechanisms

CityGML defines a rich and general-purpose information model for 3D city and landscape models. While CityGML provides a common basis for a multitude of different use cases and applications, specific applications might have modelling requirements that go beyond the predefined elements of the CityGML conceptual model.

CityGML affords this flexibility by providing a slim base model with the following distinct extension mechanisms:

1. Code lists,
2. Generic objects and attributes, and
3. Application Domain Extensions (ADEs).

These extension mechanisms have been introduced with CityGML 1.0 and are since widely used and accepted in the CityGML community.

5.1. Code Lists

When an attribute may have only one value selected from a small and fixed set of values, CityGML 3.0 specifies those as an enumeration. When the set of possible values is a priori unknown but predefined and fixed for a specific application domain or user organization, the values may be modelled as classes with the stereotype «CodeList».

Many attributes of CityGML types use a code list as data type such as, for instance, the attributes *class*, *usage,* and *function* of city objects. A code list defines a value domain including a code for each permissible value. In contrast to fixed enumerations, modifications to the value domain become possible with code lists.

The feasible values for attributes with code lists may substantially vary for different applications, uses cases, information communities or even countries (e.g., due to national law or regulations). For this reason, code lists are modelled as empty classes without predefined normative content. The governance of code lists is rather decoupled from the governance of the CityGML conceptual model, and the contents may be defined and managed outside this International Standard by any organization or information community according to their information needs. Code lists should be made available as publicly accessible resource, either globally or within an application domain.

Rules and constraints for the encoding of both code lists and attributes having a code list as value are not subject of this document but are defined in a corresponding CityGML Encoding Specification. It is recommended, though, that attributes should be encoded such that they can take a value from a code list and, optionally, provide an identifier that references the code list in a unique way.

5.2. Generic Objects and Attributes

NOTE	Fix link to generics section of the standard.

Generic objects and attributes are available from the Generics module of the CityGML Conceptual Model (cf. [rc_generics_section]). A generic object is intended to be used as proxy for a real-world feature that is not mapped by a specific class in CityGML. Depending on the type and geometric characteristics of the real-world feature, CityGML offers the proxy classes *GenericLogicalSpace*, *GenericOccupiedSpace*, *GenericUnoccupiedSpace* and *GenericThematicSurface* to capture the feature. Each proxy can have a semantic meaning defined by the attributes *class*, *function*, and *usage*.

Generic attributes are name-value pairs that can be assigned to any city object (i.e., an instance of *core:AbstractCityObject*) to augment it with application data not covered by the predefined attributes. The attribute *name* can be freely chosen to identify the piece of information represented by the generic attribute. A fixed list of simple data types is offered as possible domains for the attribute *value*. Generic attributes can be grouped into named collections using the *GenericAttributeSet* data type.

The main advantage of generic objects and attributes is that they are simple and easy-to-use to represent application-specific content. Since this extension mechanism is built into the conceptual model of CityGML, it provides the capability of ad-hoc data enrichment (“at run-time”) without the need for modifying the conceptual model. This flexibility also faces disadvantages though:

- Generic objects are “flat” and cannot be decomposed into sub-features and feature hierarchies like other CityGML features such as, for instance, buildings or transportation features. However, they may be related to other city objects through the inherited *relatedTo* association.
- Names, data types, and multiplicities of generic attributes cannot be specified in a formal way. Consequently, there is no guarantee for an application that a generic attribute of a specific name and type is available a minimum or maximum number of times for a given city object.
- Name clashes between generic attributes from different applications are possible and cannot be avoided in a formal way, which might impede semantic interoperability.
- There is only a limited number of predefined simple data types available for generic attributes.

To avoid semantic interoperability issues, generic objects and attributes shall only be used if a more specific feature class or attribute is not available from the CityGML conceptual model.

5.3. Application Domain Extension (ADE)

An *Application Domain Extension* (ADE) is a formal and systematic extension of CityGML for a specific application or domain in the form of a conceptual UML model. The application data is mapped to a set of additional classes, attributes, and relations. ADEs may use elements from CityGML, for instance, to derive application-specific subclasses, to inject additional properties, to associate application data with predefined CityGML content, or to define value domains for attributes.

The ADE mechanism allows application-specific information to be aligned with the conceptual model of CityGML in a well-structured and systematic way. By this means, CityGML can be extended to meet the information needs of an application while at the same time preserving its concepts and semantic structures. Moreover, and in contrast to generic city objects and attributes, application data can be validated against the formal definition of an ADE to ensure semantic interoperability.

Previous versions of CityGML defined the ADE mechanism solely on the level of the XML Schema

encoding. With CityGML 3.0, ADEs become platform-independent models on a conceptual level that can be mapped to multiple and different target encodings.

ADEs have successfully been implemented in practice and enable a wide range of applications and use cases based on CityGML. An overview and discussion of existing ADEs is provided in [[Biljecki2018]].

NOTE	fix uml notation section reference and Biljecki2018 citation.

5.3.1. General Rules for ADEs

An ADE shall be defined as conceptual model in UML in accordance with the conceptual modelling framework of the ISO 19100 series of International Standards and by adhering to the General Feature Model and the rules and constraints for application schemas as specified in ISO 19109 and ISO/TS 19103. The UML notations and stereotypes used in the CityGML conceptual model should also be applied to corresponding model elements in an ADE.

Every ADE shall be organized into one or more UML packages having globally unique namespaces and containing all UML model elements defined by the ADE. An ADE may additionally import and use predefined classes from external conceptual UML models such as the CityGML modules or the standardized schemas of the ISO 19100 series of International Standards.

5.3.2. Defining New ADE Model Elements

Following ISO 19109, features are the primary view of geospatial information and the core elements of application schemas. ADEs therefore typically extend CityGML by defining new feature types appropriate to the application area together with additional content such as object types, data types, code lists, and enumerations.

Every feature type in an ADE shall be derived either directly or indirectly from the CityGML root feature type *core:AbstractFeature* or, depending on its type and characteristics, from a more appropriate subclass thereof. According to the general space concept of CityGML, features representing spaces or space boundaries shall be derived either directly or indirectly from *core:AbstractSpace* or *core:AbstractSpaceBoundary* respectively. UML classes representing top-level feature types shall use the *«TopLevelFeatureType»* stereotype.

In contrast to feature types, object types and data types are not required to be derived from a predefined CityGML class unless explicitly stated otherwise.

ADE classes may have an unlimited number of attributes and associations in addition to those inherited from their parents. Attributes can be modelled with either simple or complex data types. To ensure semantic interoperability, the predefined types from CityGML or the standardized schemas of the ISO 19100 series of International Standards should be used wherever appropriate. This includes, amongst others, basic types from ISO/TS 19103, geometry and topology objects from ISO 10107, and temporal geometry and topology objects from ISO 19108.

If a predefined type is not available, ADEs can either define their own data types or import data types from external conceptual models. This explicitly includes the possibility to define new geometry types not offered by ISO 19107. Designers of an ADE should however note that software

might not be able to properly identify and consume such geometry types.

A feature type capturing a real-world feature with geometry should be derived either directly or indirectly from *core:AbstractSpace* or *core:AbstractSpaceBoundary*. By this means, the predefined spatial properties and the associated LOD concept of CityGML are inherited and available for the feature type. If, however, these superclasses are either inappropriate or lack a spatial property required to represent the feature, an ADE may define new and additional spatial properties. If such a spatial property should belong to one of the predefined LODs, then the property name shall start with the prefix "lod*X*", where *X* is to be replaced by an integer value between 0 and 3 indicating the target LOD. This enables software to derive the LOD of the geometry.

Constraints on model elements should be expressed using a formal language such as the Object Constraint Language (OCL). The ADE specifies the manner of application of constraints. However, following the CityGML conceptual model, constraints should at least be expressed on ADE subclasses of *core:AbstractSpace* to limit the types of space boundaries (i.e., instances of *core:AbstractSpaceBoundary*) that may be used to model the boundary of a space object.

NOTE	add ADE examples

Illustrative examples for ADEs can be found in the CityGML 3.0 User Guide.

5.3.3. Augmenting CityGML Feature Types with Additional ADE Properties

If a predefined CityGML feature type lacks one or more properties required for a specific application, a feasible solution is to derive a new ADE feature type as subclass of the CityGML class and to add the properties to this subclass. While conceptually clean, this approach also faces drawbacks. If multiple ADEs require additional properties for the same CityGML feature type, this will lead to many subclasses of this feature type in different ADE namespaces. Information about the same real-world feature might therefore be spread over various instances of the different feature classes in an encoding making it difficult for software to consume the feature data.

For this reason, CityGML provides a way to augment the predefined CityGML feature types with additional properties from the ADE domain without the need for subclassing. Each CityGML feature type has an extension attribute of name "adeOf*FeatureTypeName*" and type "*ADEOfFeatureTypeName*", where *FeatureTypeName* is replaced by the class name in which the attribute is defined. For example, the *bldg:Building* class offers the attribute *bldg:adeOfBuilding* of type *bldg:ADEOfBuilding*. Each of these extension attributes can occur zero to unlimited times, and the attribute types are defined as abstract and empty data types.

If an ADE augments a specific CityGML feature type with additional ADE properties, the ADE shall create a subclass of the corresponding abstract data type associated with the feature class. This subclass shall also be defined as data type using the stereotype *«DataType»*. The additional application-specific attributes and associations are then modelled as properties of the ADE subclass. This may include, amongst others, attributes with simple or complex data type, spatial properties or associations to other object and feature types from the ADE or external models such as CityGML.

The predefined "*ADEOfFeatureTypeName*" data types are called "hooks" because they are used as the head of a hierarchy of ADE subclasses attaching application-specific properties. When subclassing the "hook" of a specific CityGML feature type in an ADE, the properties defined in the

subclass can be used for that feature type as well as for all directly or indirectly derived feature types, including feature types defined in the same or another ADE.

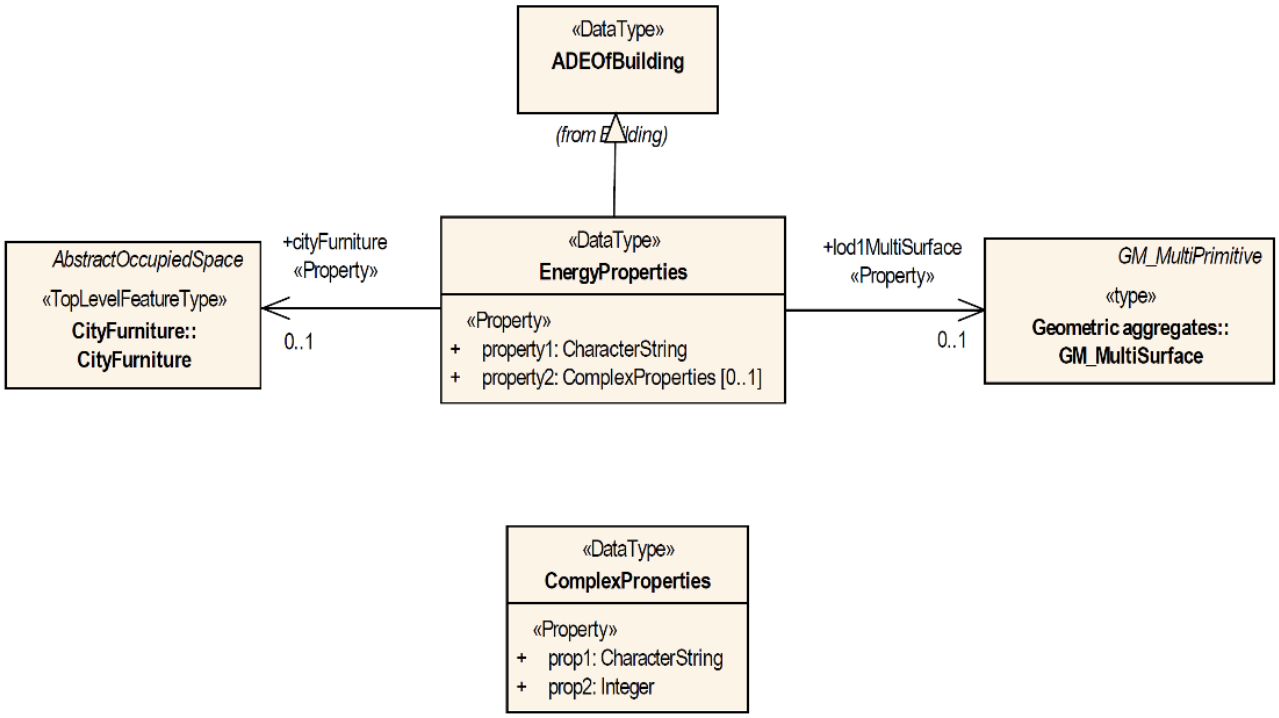
Multiple distinct ADEs can use the “hook” mechanism to define additional ADE properties for the same CityGML feature type. Since the “adeOf*FeatureTypeName*” attribute may occur multiple times, the various ADE properties can be exchanged as part of the same CityGML feature instance in an encoding. Software can therefore easily consume the default CityGML feature data plus the additional properties from the different ADEs.

Content from unknown or unsupported ADEs may be ignored by an application or service consuming an encoded CityGML model.

Designers of an ADE should favor using this “hook” mechanism over subclassing a CityGML feature type when possible. If an ADE must enable other ADEs to augment its own feature types (so-called ADE of an ADE), then it shall implement “hooks” for its feature types following the same schema and naming concept as in the CityGML conceptual model.

NOTE	Reference to the user guide.

The following UML fragment shows an example for using the "hook" mechanism. For more details on this and other example ADEs, please see the CityGML 3.0 User Guide for an example ADE.



5.3.4. Encoding of ADEs

This document only addresses the conceptual modelling of ADEs. Rules and constraints for mapping a conceptual ADE model to a target encoding are expected to be defined in a corresponding CityGML Encoding Standard. If supported, an ADE may provide additional mapping rules and constraints in conformance with a corrresponding CityGML Encoding Standard.

5.4. Core

NOTE Under Construction

5.5. Appearance

NOTE Under Construction

5.6. Bridge Model

NOTE Under Construction

5.7. Building Model

NOTE Under Construction

5.8. City Furniture

NOTE Under Construction

5.9. City Object Group

NOTE Under Construction

5.10. Construction

NOTE Under Construction

5.11. Dynamizer

NOTE Under Construction

5.12. Generics

NOTE Under Construction

5.13. Land Use

NOTE Under Construction

5.14. Point Cloud

NOTE Under Construction

5.15. Relief

NOTE Under Construction

5.16. Transportation

NOTE Under Construction

5.17. Tunnel Model

NOTE Under Construction

5.18. Vegetation

NOTE Under Construction

5.19. Versioning

NOTE Under Construction

5.20. Waterbodies

NOTE Under Construction

Annex A: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version

Annex B: Bibliography

NOTE

Example Bibliography (Delete this note).

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

[1] OGC: OGC Testbed 12 Annex B: Architecture. (2015).