

# MODELING AN APPLICATION DOMAIN EXTENSION OF CITYGML IN UML

Linda van den Brink<sup>1</sup>, Jantien Stoter<sup>1,2,3</sup>, Sisi Zlatanova<sup>2</sup>

<sup>1</sup>Geonovum, Amersfoort, {l.vandenbrink|j.stoter}@geonovum.nl

<sup>2</sup>OTB, TU Delft, {j.e.stoter|s.zlatanova}@tudelft.nl

<sup>3</sup>Kadaster, Apeldoorn, Jantien.stoter@kadaster.nl

ISPRS Commission IV, WG IV/8

**KEY WORDS:** three-dimensional, modeling, abstraction, structure

## ABSTRACT:

This paper presents key aspects of the development of a Dutch 3D standard IMGeo as a CityGML ADE. The new ADE is modeled using UML class diagrams. However the OGC CityGML specification does not provide clear rules on modeling an ADE in UML. This paper describes how the extension was built, which provides general insight how CityGML can be extended for a specific applications starting from the UML diagrams. Several alternatives for modeling ADEs in UML have been investigated and compared. The best suited for the 3D standard option is selected and applied. Open issues and challenges are discussed in the conclusions.

## 1. INTRODUCTION

Recently a national 3D standard has been established in The Netherlands as a CityGML ADE (Van den Brink et al 2012; Stoter et al 2011). This ADE completely integrates CityGML with the existing national Information Model for large scale topography (called Information Model Geography or 'IMGeo'). IMGeo is modeled in UML (Unified Modeling Language), and contains object definitions for large scale representations of roads, water, land use/land cover, bridges, tunnels etc. and their properties, and prescribes 2D point, curve or surface geometry for all objects. As the new version of IMGeo (version 2.0) is completely integrated with CityGML, 2D IMGeo data can be extended into 2.5D (i.e. as height surface representation) and 3D (as volumetric representation) according to geometric and semantic principles of CityGML.

In line with the Dutch practice of modeling geo-information in UML the IMGeo CityGML ADE is modeled using UML class diagrams. However, the CityGML specifications do not provide rules or guidance on modeling an ADE in UML (CityGML ADE, 2012). It describes only how an ADE must be modeled in the XML schemas.

Based on the lessons learned from developing the CityGML-IMGeo ADE, this paper describes how CityGML can be extended for specific applications starting from the UML diagrams. A complete description of the CityGML-IMGeo ADE can be found in Van den Brink et al (2011). This paper summarizes the technical modeling, i.e. how the UML models of CityGML can be extended to also support the concepts defined in a specific domain, and how a GML application schema (OGC 2007) conforming to the CityGML ADE rules can be automatically generated from the UML model. Our experiences can serve as best practice to standardize the developments of domain specific CityGML ADEs in the near future. Firstly, because this process is not standardized yet and, secondly, because our approach was established through intensive discussions on possible alternatives with the OGC CityGML Working Group, members of the Special Interest Group 3D (SIG 3D) and other experts. An important step in this process, has been the decision of SIG 3D to adopt our UML modeling approach for ADE's in March, 2012.

The paper is organized as follows. Section 2 describes the optimal modeling approach, which was selected from several alternatives. Section 3 explains how the selected modeling approach has been applied to model the CityGML ADE 'IMGeo 2.0' and Section 4 concludes on findings and topics for further research.

## 2. APPROACH FOR MODELING CITYGML ADES IN UML

This section presents the selection process of the optimal approach for modeling application domain extensions of CityGML in UML. During the development of CityGML-IMGeo, intensive discussions took place on possible alternatives for modeling the ADE in UML, specifically with the OGC CityGML Working Group, members of the Special Interest Group 3D (SIG 3D) and other experts. Some of the alternatives were based on having only generic extension placeholders in the UML model, while the extension would be described informally in some way outside the model. In other alternatives the extension would be described in UML. Most of these alternatives were in some degree in conflict with rules of UML, the ISO19109 General Feature model, or the modular specification standard of OGC (OGC, 2009).

After comparing the advantages and disadvantages of the alternatives, one alternative was selected as the best option for the IMGeo ADE. This approach defines the properties to be added in subclasses in the ADE package but suppresses these subclasses from the generated XML Schema.

An extension subclass is marked as ADE extension in the UML model using a stereotype or tagged value. A stereotype is preferred because it makes clear from the UML diagrams that the ADE subtype is not mapped to an XML Schema component. Tagged values are not always (and usually not) shown in the graphical notation. However, this could be viewed as violating the GML encoding rule that stereotypes are used for conceptual aspects and tagged values for encoding-related aspects.

Several disadvantages to this approach were noted by the participants of the discussions. 1) It is confusing to introduce an

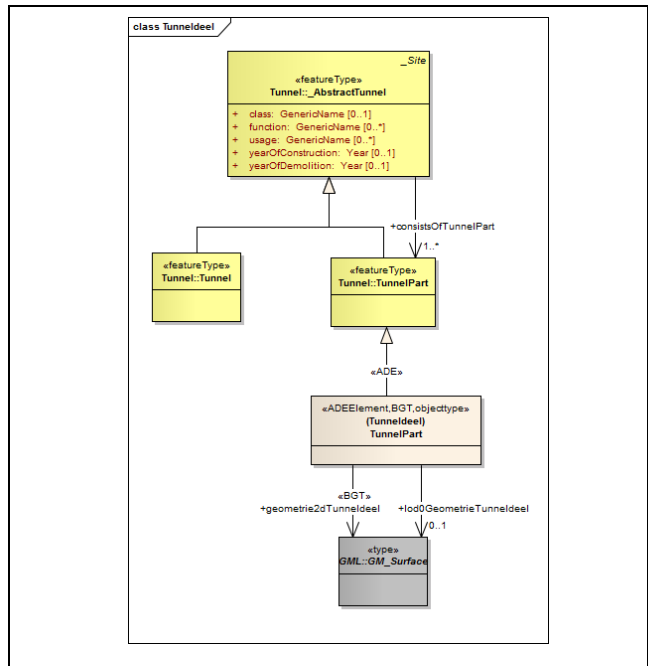
ADE subclass of a CityGML type although the ADE hooks provide a means to avoid subclassing. The ADE hooks represent a concept of attribute substitution, for which there is no standard expression method in UML. In UML, using a subclass for extending another class is appropriate. 2) In UML a subclass inherits all methods and attributes from its super class, but in this case that is not intended. In order to stress this, the generalization relation between subclass and CityGML superclass receives a stereotype <<ADE>>. 3) Because in UML the extension is modelled with subclasses, it is not possible to create an instance diagram where an object instance is shown combining properties from different ADE's. The only way to show instance data is on the XML level.

The fact that in the XML Schema implementation the subclasses are omitted, is seen as a technical implementation choice to allow the combining of properties from different ADEs. While this is a valid reason on the technical level, it is not taken to mean that in the conceptual UML model subclassing should also be avoided.

This section explains how the selected modeling approach was applied to model the CityGML ADE IMGeo 2.0. Although the focus in this section is on realising a CityGML ADE for the Dutch information model “IMGeo”, the followed procedure contains generalities that can be used to model other ADEs in UML as well.

Since CityGML is only available as xml schema, the first step is to recreate the UML model in the modeling tool Enterprise Architect, based on (OGC 2008). In the next step all IMGeo classes are modeled as subclasses of CityGML classes. Using the selected modeling approach, these subclasses get the same class name as the CityGML class they are extending. The stereotype <<ADEElement>> is assigned to the subclasses. This stereotype was proposed by Benner and Haeefe during the discussion on the selected modeling alternative. This stereotype marks these classes as subtypes that only add properties to the CityGML class, and accordingly no XML component for these classes will be created in the XML Schema. In addition, the specialization relation with the CityGML class is marked with a stereotype <<ADE>>. For documentation purposes, a Dutch translation of the subclass name is added as an alias.

LOD0 geometry properties). The yellow classes are classes from the CityGML Tunnel package. The <<ADEElement>> TunnelPart is a class defined in the IMGeo ADE package as a subclass of CityGML TunnelPart class. The Dutch alias is shown between brackets on the class diagram.



To identify equivalent concepts that can be modeled via this subclassing method, first a conceptual mapping was made between CityGML and the application information model IMGeo. These mappings compared the concepts at semantic level, i.e. independent of which LOD the concept appears in CityGML.

If it is not possible to remodel the concept into a CityGML class, CityGML is extended with a new class, as a subclass of one of the CityGML classes. Figure 2 shows an example in the IMGeo ADE of a class which is not available in CityGML but needed in IMGeo. The class ‘*OverigeConstructie*’ (OtherConstruction) is a class to represent man-made constructions other than buildings, bridges and tunnels. Examples are water management constructs such as pumping plants, locks, and weirs but also wharfs, fences, loose-standing walls, high-tension line towers and wind turbines. It is modeled as a <<featureType>> subclass of the CityGML class \_Site (with a Dutch class name) which is not suppressed from the

XML Schema. The class has its own properties which are modeled similar to CityGML classes, such as implicit geometry on different LODs as well as 2D and 3D geometry up to LOD3.

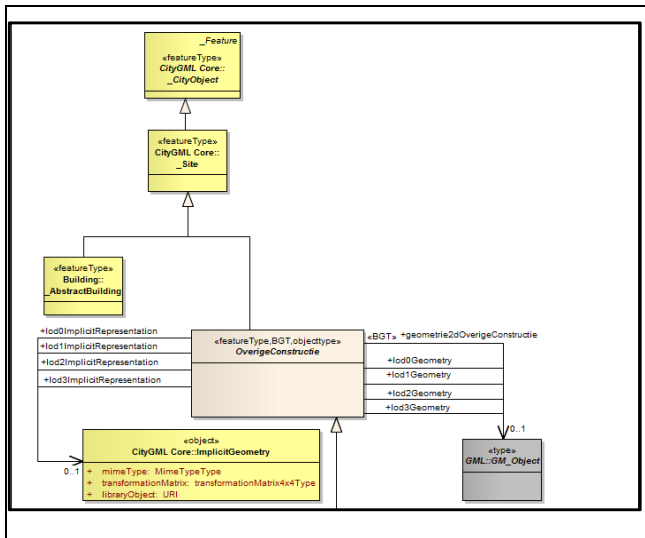


Figure 2: OverigeConstructie, new class added, derived from \_Site

### 3.2 The use of code lists in the ADE

CityGML provides code lists to allow predefined values for the CityGML attributes. However, the CityGML-IMGeo ADE makes use of national classification code lists instead of the CityGML code lists. Several aspects gave reasons to do this (this is in line with the findings of Portele 2009): the national lists are specifically suited to the Dutch context and contain a definition for each concept, approved by the Dutch organizations involved. Other reasons for not using the CityGML code lists are that IMGeo favors Dutch language code lists and that the CityGML standard does not provide definitions for the code list values, which makes it hard to decide which value to use. There is no need to map the Dutch code lists to the CityGML code lists, as these are non-normative and software does not check on code list values nor process them in specific ways.

Both CityGML and GML do not provide a normative way to structure code lists. Prominent choices are GML dictionary and SKOS (Simple Knowledge Organisation System; SKOS, 2012). GML dictionary was considered but not selected, because these are deprecated in GML 3.3, while SKOS adoption is growing in the geo community. SKOS was therefore selected.

The code lists are maintained in the UML model and XML structured code lists can be generated from the UML using a ShapeChange customization which allows generation of SKOS-encoded code lists from UML classes with a <<codeList>> stereotype. The disadvantage of maintaining the code lists in the UML model, is that the UML model needs to be updated in case the code lists need an update. For IMGeo the code lists are considered as part of the standard and allowed to change only when a new IMGeo version is published.

The SKOS code lists will be published in a national, publicly available registry, which also contains the IMGeo XML Schema. Each code list and code list value is accessible via its own URL.

Code list validation can be done using standard XML techniques such as Schematron constraints (ISO/IEC 19757-3:2006(E)). Further work is needed to assess how the IMGeo code lists are best represented and structured in SKOS.

### 3.3 Generating XML Schema from the UML ADE

The Java tool ShapeChange is used to generate an XML Schema (GML application schema) from the ADE defined in UML (Portele, 2008). As mentioned before, ShapeChange implements the UML to GML encoding rules described in ISO 19136, ISO 10118, and ISO 19109. ShapeChange was modified to add a custom encoding rule for classes with the <<ADEElement>> stereotype. These classes are suppressed from the GML Application schema, while their properties are added to the ADE namespace as substitutes for the CityGML “\_GenericApplicationPropertyOf<Featuretypename>” hooks as described in CityGML 10.13.1.

For the national 3D standard it is required to identify the reference system (x, y, z) to be used in the GML files.

IMGeo 2.0 compliant test data has been generated to show how the model works when applied to data, see Figure 3.

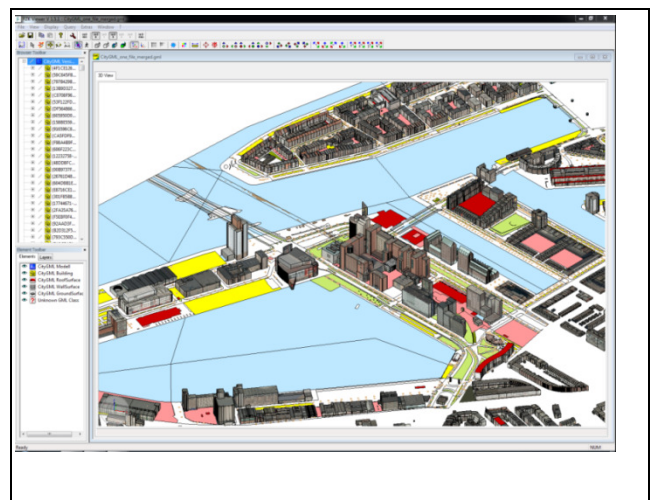


Figure 3: Visualisation of CityGML-IMGeo encoded data: CityGML LOD2

## 4. CONCLUSION AND FURTHER RESEARCH

This paper presents an approach for automatic generation of a CityGML ADE starting from UML schema. After some investigations of the several possibilities, the most beneficial for the Dutch purposes was applied for the CityGML ADE IMGeo (i.e. the Dutch national 3D standard). The main principle of the selected approach is that all classes of IMGeo are modeled as subclasses of CityGML classes and these subclasses get the same class name as the CityGML class they are extending. The stereotype <<ADEElement>> marks these classes as subtypes that only add properties to the CityGML class, and accordingly no XML component for these classes will be created in the XML Schema.

In the development of CityGML ADE IMGeo 2.0 a number of topics are identified that requires further research. Firstly, more research is needed to understand how this model works in practice including the consequences of this new modeling

method for IMGeo when used for both 2D and 3D datasets, e.g. how to preserve the links between the different LODs and how to upgrade 2D LOD to higher LODs. Secondly, knowledge is required on the ability to use 3D IMGeo data in CityGML-aware software, i.e. whether software systems are compatible with our extensions and which adaption is required. Furthermore research is needed to assess how the IMGeo code lists are best represented and structured in SKOS. Finally, the creation and management of CityGML-IMGeo data needs more research attention. Which methods can be used to generate CityGML-IMGeo data? How should this data be validated and maintained? How can 2.5D topology be created and maintained?

These open issues are currently being studied in a follow-up project of the 3D Pilot NL (Stoter et al, 2012). The first phase finished in June, 2011 and has been reported in Stoter et al (2011). Since October 2011 over 100 organizations (Geonovum, 2012) are contributing to the six activities of the second phase of the 3D Pilot NL. The activities related to learn more about the UML modeling approach for ADEs are the generation of 3D IMGeo example data (several levels of detail and several classes) and the design and implementation of a 3D validator that tests whether both the semantics and the geometry of the data are compliant with the standard.

This is the first study on extending the UML diagrams of CityGML for specific domains. Since the OGC CityGML specifications do not provide rules or guidance on correctly modeling an ADE in UML, this paper may serve as best practice for future ADEs to be modeled in UML.

## 5. ACKNOWLEDGMENTS

The authors express their sincere gratitude to Thomas Kolbe, Claus Nagel (both TU Berlin); Carsten Roensdorf (Ordnance Survey and chair of OGC CityGML working group) and Clemens Portele (interactive instruments GmbH), who actively participated in the discussions.

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs, Agriculture and Innovation (project code: 11300).

## 6. REFERENCES

CityGML ADE, 2012, current ADE are maintained at [www.citygmlwiki.org/index.php/CityGML-ADEs](http://www.citygmlwiki.org/index.php/CityGML-ADEs)

Geonovum, 2012, list of participants of second phase 3D Pilot, online, [www.geonovum.nl/dossiers/3d-pilot/deelnemersvervolg](http://www.geonovum.nl/dossiers/3d-pilot/deelnemersvervolg)

IMGeo, 2007. Informatiemodel Geografie, oct 2007 [online]. Available from [www.geonovum.nl/sites/default/files/IMGeo\\_rapport\\_definitief\\_versie\\_1.0.pdf](http://www.geonovum.nl/sites/default/files/IMGeo_rapport_definitief_versie_1.0.pdf)

ISO/IEC, 2006, 19757-3:2006(E), Information technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation – Schematron. First edition 2006-06-0

OGC, 2007, OpenGIS® Geography Markup Language (GML) Encoding Standard. Version 3.2.1, doc # OGC 07-036, [http://portal.opengeospatial.org/files/?artifact\\_id=20509](http://portal.opengeospatial.org/files/?artifact_id=20509)

OGC, 2008, OpenGIS® City Geography Markup Language (CityGML) Encoding Standard, version 1.0.0, document # 08-007r1. [http://portal.opengeospatial.org/files/?artifact\\_id=28802](http://portal.opengeospatial.org/files/?artifact_id=28802)

OGC, 2009, The Specification Model — A Standard for Modular specifications. version 1.0.0, document # 08-131r3. [https://portal.opengeospatial.org/files/?artifact\\_id=34762](https://portal.opengeospatial.org/files/?artifact_id=34762)

Portele, C., 2008, Mapping UML to GML Application Schemas; ShapeChange - Architecture and Description, version 1.0rc., [www.interactive-instruments.de/fileadmin/gdi/docs/ugas/ShapeChange-1.0.pdf](http://www.interactive-instruments.de/fileadmin/gdi/docs/ugas/ShapeChange-1.0.pdf)

Portele, C., 2009. OGC® OWS-6 UTDS-CityGML Implementation Profile, editor: Clemens Portele, version 0.3.0, OGC 09-037r1

SKOS, 2012, [online] Available from [www.w3.org/2004/02/skos/](http://www.w3.org/2004/02/skos/)

Stoter, J., Vosselman, G., Goos, J., Zlatanova, S., Verbree, E., Klooster, R. and Reuvers, M., 2011. Towards a National 3D Spatial Data Infrastructure: Case of The Netherlands. PFG Photogrammetrie, Fernerkundung, Geoinformation, 2011(6): 405-420.

Stoter, J.E., Jacob Beetz, Hugo Ledoux, Marcel Reuvers, Rick Klooster, Paul Janssen, Friso Penninga, Sisi Zlatanova, Lida van den Brink, 2012, Implementation of a national 3D standard: case of The Netherlands, 3D GeoInfo Symposium, Quebec, Canada, 16-17 May. To be published in Lecture Notes in Geoinformation and Cartography (Springer).

Van den Brink, L., J.E. Stoter and S. Zlatanova, 2012. Establishing a national standard compliant to CityGML, accepted for *International Journal of Geographical Information Science IJGIS*