Delft University of Technology

Harmonising the OGC Standards for the Built Environment
A CityGML Extension for LandInfra

Kavisha, Kavisha; Labetski, Anna; Arroyo Ohori, Ken; Ledoux, Hugo; Stoter, Jantien

DOI
10.3390/ijgi8060246

Publication date
2019

Document Version
Final published version

Published in
ISPRS International Journal of Geo-Information

Citation (APA)
Kavisha, K., Labetski, A., Arroyo Ohori, K., Ledoux, H., & Stoter, J. (2019). Harmonising the OGC Standards for the Built Environment: A CityGML Extension for LandInfra . *ISPRS International Journal of Geo-Information*, *8*(6), Article 246. https://doi.org/10.3390/ijgi8060246

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

International Journal of
Geo-Information

MDPI

Article

Harmonising the OGC Standards for the Built Environment: A CityGML Extension for LandInfra

Kavisha Kumar *,†, **Anna Labetski** †, **Ken Arroyo Ohori** †, **Hugo Ledoux** † **and Jantien Stoter** †

3D Geoinformation, Delft University of Technology, Julianalaan 134, 2628 BL Delft, The Netherlands; a.labetski@tudelft.nl (A.L.); g.a.k.arroyoohori@tudelft.nl (K.A.O.); h.ledoux@tudelft.nl (H.L.); j.e.stoter@tudelft.nl (J.S.)

* Correspondence: k.kavisha@tudelft.nl

† These authors contributed equally to this work.

Received: 8 April 2019; Accepted: 26 May 2019; Published: 29 May 2019

Abstract: The relatively new Open Geospatial Consortium (OGC) standard LandInfra documents in its data model land and civil engineering infrastructure features. It has a Geography Markup Language (GML) implementation, OGC InfraGML, which has essentially no software support and is rarely used in practice. In order to share the benefits of LandInfra (and InfraGML) with a wider public, we have created the Infra Application Domain Extension (ADE), a CityGML ADE that allows us to store LandInfra features in CityGML. In this paper, we semantically map LandInfra to CityGML, describe our ADE, and discuss a few used cases where our ADE can be useful for applications for the built environment. We also provide software to automatically convert datasets from InfraGML to CityGML (and our ADE), and vice versa, as well as to validate them, which will help practitioners generate real-world InfraGML datasets.

Keywords: Infra ADE; LandInfra; InfraGML; CityGML; GIS; 3D city models

1. Introduction

LandInfra [1] is a relatively new Open Geospatial Consortium (OGC) open standard for land and infrastructure features, integrating concepts from Building Information Modelling (BIM) and Geographical Information Systems (GIS). Thus it partially overlaps the main standard of the 3D GIS world, CityGML, such as in its thematic classes `Building`, `Road` and `Railway` (`Transportation` in CityGML), and `LandSurface` (`ReliefFeature` in CityGML) [1]. However, LandInfra is much more powerful in some areas, as it has a more detailed representation for land and infrastructure features, e.g., administrative units, ownership rights, spatial units for land use (land parcels and the legal spaces of buildings), surveying and representation, alignment for roads and railways, subsurface models for terrain, etc.

LandInfra has a Geography Markup Language (GML) implementation: InfraGML, which is also an OGC standard. However, it has no software support yet and is barely used in practice, which means that the advantages of LandInfra (and InfraGML) are not yet being used. The fact that a standard has been tested and implemented in code is a positive feature of the standardization approach, which increases the usability of the standard. After all, we do not want LandInfra, a connecting link between BIM and GIS, to suffer because of low support and no reference implementation. Therefore, in order to encourage the adoption of LandInfra's features, we have developed the Infra Application Domain Extension (ADE)—an ADE for CityGML that integrates LandInfra's concepts into CityGML, which we describe in this paper including the steps that we have taken and decisions we have made to develop the ADE. The idea behind the integration is to take the best of both worlds (i.e., CityGML and LandInfra) and have more information than CityGML for specific applications or uses, such as urban environment analysis, subsurface modelling etc.

First, we provide a brief review of the data models of LandInfra and CityGML in Section 2. Second, we provide a complete mapping between LandInfra and CityGML in Section 3, where we identify the matching classes and attributes in the two data models, as well as the LandInfra classes and attributes that do not have a semantic equivalent in CityGML but are useful for the built environment applications, e.g., the material of a building, and the life cycle phase of a building. Third, we describe our CityGML Infra ADE in Section 4, where the missing LandInfra concepts are added to CityGML. Fourth, we provide a few uses in Section 6 for the CityGML Infra ADE to demonstrate the benefits of our ADE in practice.

As a proof of concept of our Infra ADE, we have also implemented two software prototypes to convert datasets from InfraGML to CityGML (and our ADE) and vice versa, as well as a prototype to ensure that the InfraGML files are valid. These are described in Section 5. Finally, we close the article with conclusions and future work.

2. Background

2.1. CityGML

CityGML [2] is an open standard from the OGC for the storage and exchange of 3D city models, including their geometry, semantics, and graphical appearance. It is implemented as an application schema of the GML version 3.1.1 [3].

The data model of CityGML is comprised of a core module and several thematic modules such as `Building`, `Relief`, `Bridge`, `Transportation`, `Vegetation`, and `WaterBody`, which include various types of city objects and their associated semantic properties. Moreover, objects that are not explicitly included in these modules, such as pipes and road noise barriers, can be stored by extending the data model using either of two mechanisms: Generics or ADEs [4–6].

Generics allow one to easily extend the city objects in CityGML with a `GenericCityObject` and attributes (`_genericAttribute`) without making any changes in the CityGML schema. However, Generics have limitations. CityGML datasets with generic objects and attributes cannot be fully validated against the schema because there is no formal specification of their names and data types. Moreover, name conflicts of the generic objects and attributes may occur. This limits semantic and syntactic interoperability when using Generics.

The second approach is Application Domain Extensions (ADEs), which is more complex but also more structured than Generics. ADEs are formally specified in a separate XML Schema Definition (XSD) file and have their own namespace. Because of this, datasets that use ADEs can be semantically validated. ADEs can be modelled in two ways: First, directly in the XSD schema file; second, by extending the Unified Modelling Language (UML) model of CityGML with application-specific attributes/objects, and later generating the XML schema from the UML model [4–6]. ADEs do not need a formal approval by any standardisation body and can be developed by anyone. They are actively used to create application-specific extensions, such as the Noise ADE for noise mapping [2,7], the Energy ADE for energy modelling [8], the Dutch IMGeo ADE [4], the iTINs ADE for handling massive terrains [5], the Metadata ADE [9], etc. See Biljecki et al. [6] for an overview of existing ADEs.

2.2. LandInfra and InfraGML

LandInfra [1] is also an open standard from the OGC, but one which focuses on land and civil engineering infrastructure facilities, including roads, buildings, railways, projects, alignments, surveys, land features and land divisions. 'Wet' infrastructure is slated for a future version, including features such as storm drainage, waste water, and water distribution systems. LandInfra was introduced as the proposed successor to LandXML [10].

LandInfra has 10 main requirements classes (summarised in Table 1). The self-named `LandInfra` is the only mandatory class. InfraGML is the GML based implementation of LandInfra, which is published as an eight part OGC standard: LandInfra Core (Part 0), Land Features (Part 1), Facilities and Projects (Part 2), Alignments (Part 3), Roads (Part 4), Railways (Part 5), Survey (Part 6), and Land Division (Part 7). Each part has a separate schema (XSD file).

Table 1. Main LandInfra requirements classes.

#	Class	Summary	InfraGML Part
1	`LandInfra`	Mandatory core with dataset information and common types	0
2	`Facility`	Collection of buildings, civil engineering works and their siteworks	2
3	`Project`	Activity related to the improvement of a facility	2
4	`Alignment`	Positioning element for locating physical elements	3
5	`Road`	Roads with 3D elements	4
6	`Railway`	3D railway elements and track geometry	5
7	`Survey`	Information related to surveys, e.g., equipment, results, etc.	6
8	`LandFeature`	Whether natural or man-made, in the surface or subsurface	1
9	`LandDivision`	Public (political, judicial, or executive) or private land divisions	7
10	`Condominium`	Ownership of private and public units in a multi-unit building	7

Given that LandInfra is quite a young standard, it is currently difficult to identify any solid examples of its usage in practice. The vast majority of academic articles that mention LandInfra only describe the possibility of utilising it for various applications as future work.

Among those papers, the majority of them discuss the relationship between LandInfra and the ISO 19152 LADM (Land Administration Domain Model) [11–16]. There are also several papers examining the applicability of LandInfra for extending the transportation modelling of roads [17] in the context of road asset management [18–20], and for the interoperability between the LandInfra and RailTopoModel [21] for railway infrastructure [22]. Others discuss the usability of LandInfra (& InfraGML) in specific use cases, such as in the creation of a tax valuation information model [15], underground utility network modelling [23] and possible alignments [24] with the CityGML Utility Networks ADE [25] and PipelineML [26], environmental acoustic studies [27], and modelling legal interests and legal boundaries [28].

Furthermore, attempts are being made to align IFC (Industry Foundation Classes) and LandInfra. For instance, the LandInfra `Alignment` requirement class is based on the buildingSMART IFC Alignment 1.0 standard [1]. It was developed jointly by the OGC and the buildingSMART Infrastructure IFC Alignment project team to ensure interoperability between the two standards in the future. Moreover, buildingSMART is currently working on an IFC Infrastructure extension to model the spatial and physical components of the roads, bridges, and other structures in IFC [29] so that the forthcoming IFC conceptual model for roads and railways be compatible with the LandInfra and InfraGML.

3. Methodology

Mapping between LandInfra and CityGML

LandInfra and CityGML have significant similarities and differences, which we have discussed in detail in Kumar et al. [30]. After comparing the two standards and analysing the individual correspondences of the classes, attributes and other concepts in the data model of LandInfra to their equivalent ones in CityGML, we found that they fit into five different categories as mentioned below. To avoid any confusion in the names of the classes, the LandInfra and CityGML classes are appended with prefixes `LI` and `CG` respectively, in this paper.

1. Classes (and their attributes) which can be directly mapped from LandInfra to CityGML, e.g., `LI::LandSurface` with `CG::TINRelief`, `LI::Road` with `CG::Road`, and so on.

2. LandInfra classes that require a specific attribute value to determine their corresponding matching classes in CityGML e.g., `LI::LandElement` can be mapped to `CG::PlantCover`, `CG::WaterBody` or `CG::TINRelief` based on the value of its attribute `elementType`. The values for `elementType` attribute are defined in the LandInfra codelist `LandElementType`.
3. LandInfra classes (and their attributes) that do not have a semantically equivalent class in CityGML e.g., classes such as *`LI::_LandLayer`* and `LI::Facility`.
4. Classes (and their attributes) that define relationships among features e.g., how different LandInfra features (`LI::Feature(s)`) are related to each other via `LI::FeatureAssociation`, which are not present in CityGML.
5. LandInfra constraints such as data types, enumerations, and codelists which do not have a semantic equivalent in CityGML.

Based on these categories, we have developed a complete mapping from LandInfra to CityGML, which is presented in Table 2. Notice that many classes in LandInfra do not have clear correspondences in CityGML, which is largely because LandInfra is much more detailed than CityGML with respect to the semantic information and relationships of land and infrastructure features. Loss of information while converting InfraGML datasets to CityGML is thus inevitable without an extension to the CityGML data model.

Table 2. Mapping between the LandInfra conceptual model and CityGML. For simplicity, we only show here the mapping for all the LandInfra main requirement classes (marked with M) and a few other LandInfra classes as examples.

#	LandInfra	CityGML	Description
1	`LandInfraDataset` (M)	`CityModel`	Aggregations of features with optional metadata.
2	`Feature`	*`_CityObject`*	The base classes for all the features. Both are derived from GML class *`_Feature`*, but `LI::Feature` is a concrete class whereas *`CG::_CityObject`* is abstract; they are only conceptually similar.
3	`Document`	—	No matching semantic class is available.
4	`SurveyMark`	—	No matching semantic class is available.
5	`Project` (M)	—	No matching semantic class is available.
6	`Facility` (M)	*`_Site`*	Both include buildings and other civil engineering structures, such as tunnels and bridges, but `LI::Facility` also defines runways, pipelines and water systems, which are not present in CityGML. However, `LI::Facility` is a concrete class whereas *`CG::_Site`* is abstract; they are only conceptually similar.
7	`FacilityPart`	—	No matching semantic class is available.
8	`Building`	`Building`	In LandInfra it is a subclass of `LI::FacilityPart`, whereas in CityGML it is a part of the `CG::Building` module.
9	`LandFeature` (M)	—	No matching semantic class is available.

10	`LandElement`	`WaterBody/`
		`PlantCover/`
		`TINRelief`
			The correspondence depends on the value of a `LI::LandElement`'s `elementType`, which can be `waterBody`, `vegetation` or `landForm`, as defined in the codelist `LandElementType`.

11	`LandSurface`	`TINRelief`	`LI::LandSurface` models terrain as a TIN, much like `CG::TINRelief`.

Table 2. *Cont.*

#	LandInfra	CityGML	Description
12	_*LandLayer*	—	No matching semantic class is available.
13	Alignment (M)	—	No matching semantic class is available.
14	Road (M)	Road	In the Transportation module in CityGML and in the Facility requirement class in LandInfra. CityGML focuses on polygon representations with the possibility to include lines as well, while LandInfra focuses on lines and surfaces (i.e., TINs).
15	Railway (M)	Railway	In the Transportation module in CityGML and in the Facility requirement class in LandInfra. LandInfra provides many more components that can be modelled.
16	LandDivision (M)	LandUse	CG::LandUse represents divisions according to specific land uses, whereas LI::LandDivision has richer semantics and is further divided according to political, judicial, or executive views, ownership, rights, and so on.
17	CondominiumBuilding (M)	—	No matching semantic class is available.
18	Survey (M)	—	No matching semantic class is available.

4. Results

4.1. The CityGML Infra ADE

In order to support LandInfra concepts in CityGML, we have developed the Infra ADE, which is able to store and manage LandInfra/InfraGML datasets in CityGML with full compatibility. We have implemented the ADE and we provide the UML model, the XSD schema, the documentation and a prototype software, which are all publicly available in our GitHub repository: https://github.com/tudelft3d/city2InfraGML.

Depending on the five cases of the classification in our analysis, we have built the Infra ADE for CityGML by:

1. Adding the missing LandInfra attributes to the CityGML classes that matched LandInfra classes (Cases 1 and 2);
2. Adding new types that represent the LandInfra classes that do not have matching CityGML classes (Case 3 and 4);
3. Adding support for the LandInfra geometry types, data types and codelists (Case 5).

These solutions are individually presented in the following subsections. Note that in order to avoid any conflict with the existing CityGML elements, the new Infra ADE elements are defined in a different namespace https://3d.bk.tudelft.nl/schemas/infraADE with an identifier 'infra'.

4.1.1. Extending the CityGML Classes That Match LandInfra Classes

CG::CityModel is extended to include LI::LandInfraDataset's classes and their specific attributes (Figure 1). It stores:

- The ID and the scope of the dataset (<infra:datasetID>);
- The metadata about the dataset (e.g., <infra:name>, <infra:dateTime> and <infra:author>);
- The associations between the features in the dataset (<infra:featureAssociation>);
- The information about the survey(s) stored in the dataset (<infra:survey>);

- The collective information about the features belonging to a particular type or authority in the dataset (`<infra:set>`).

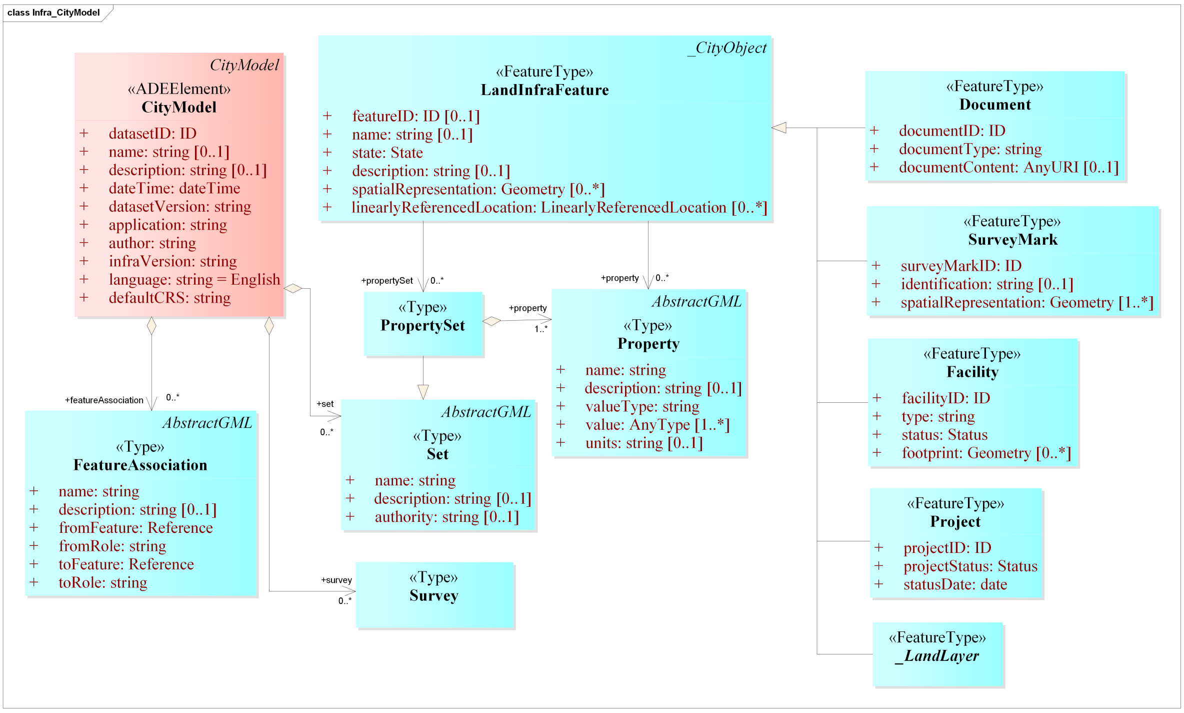


Figure 1. The Unified Modelling Language (UML) excerpt depicts extended City Geography Markup Language (CityGML) class `CG::CityModel` (marked with stereotype «ADEElement») to include attributes from LandInfra class `LI::LandInfraDataset`. The LandInfra classes such as `Facility`, `Project`, `LandFeature`, `Document`, `SurveyMark`, and `_LandLayer`, which do not have a corresponding match in CityGML are introduced in the Infra Application Domain Extension (ADE).

`CG::Railway` is extended to include `LI::Railway`'s classes and their specific attributes (Figure 2). It stores:

- The ID and the scope of the railway features present in the dataset (`<infra:railwayID>`);
- The attribute indicating if the railway feature is existing or proposed (`<infra:railwayState>`);
- The status to indicate where the railway feature is within its life cycle (`<infra:railwayStatus>`);
- The railway elements such as switches, rails, etc. present in the dataset (`<infra:railwayElement>`);
- The specifications of the cant (also called the superelevation) of the railway tracks present in the dataset (`<infra:cantSpecification>`);
- The alignments (positioning elements) used to define the geometry of the railway tracks (`<infra:railwayAlignment>`).

`CG::Road` is extended to include `LI::Road`'s classes and their specific attributes (Figure 3). It stores:

- The ID and the scope of the road features present in the dataset (`<infra:roadID>`);
- The estimated width of the road (`<infra:approximateWidth>`);
- The material of the road (`<infra:material>`);
- The attribute indicating if the road feature is existing or proposed (`<infra:roadState>`);
- The status to indicate where the road feature is within its life cycle (`<infra:roadStatus>`);
- The road elements such as pavements, side walks, etc. present in the dataset (`<infra:roadElement>`);

- The alignments (positioning elements) used to define the geometry of the roads (`<infra:roadAlignment>`);
- Alternative ways for representing a road from design perspective such as 3D StringLines (`<infra:stringLine>` aka profile views, longitudinal breaklines, long sections), and 3D surfaces (`<infra:surface>`), or as well as collections of these (`<infra:stringLineSet>` or `<infra:surfaceSet>`) [1];
- The 2D cross section views cut across the road at a particular location along the length of the road (`<infra:roadCrossSection>`).

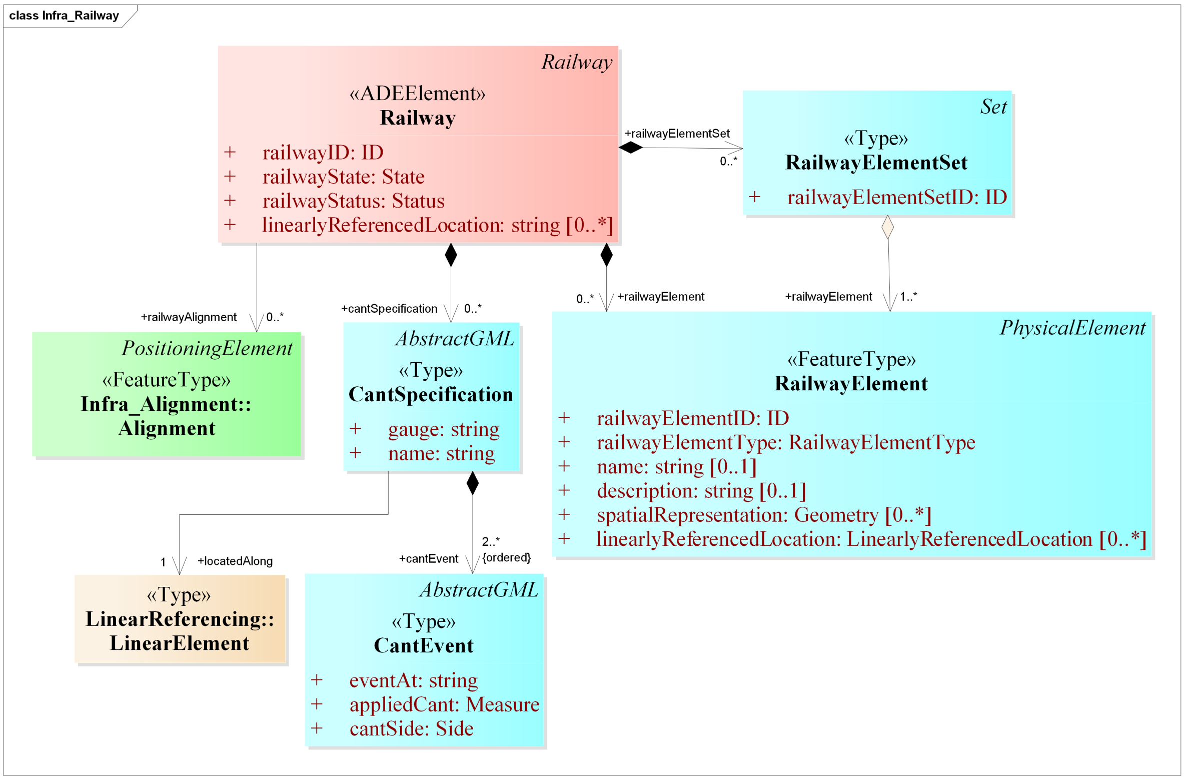


Figure 2. The UML excerpt depicts existing CityGML class `CG::Railway` (marked with stereotype «ADEElement») extended to include attributes and classes from LandInfra `LI::Railway` class in the CityGML Infra ADE. Other classes (taken from LandInfra) such as `CrossSection`, `Alignment`, etc. are also introduced.

`CG::LandUse` is extended to include `LI::LandDivision`'s classes and their specific attributes (Figure 4). It stores:

- The ID and the scope of the land division features present in the dataset;
- The type of land division. It can be public (`infra:administrativeDivision`) or private (*infra:easement* or `infra:propertyUnit`) in nature;
- The statement document which specifies which establishment or acquisition of the land (`infra:documentation`);
- The ownership rights of properties (`infra:ownership`);
- The cadastral parcels present in the dataset (`infra:landParcel`);
- The spatial units to define the geometry (shape and location) of the land parcels, easements, and other administrative divisions (`infra:SpatialUnit`);
- The bounding elements to specify the boundary of the spatial units (`infra:boundingElement`).

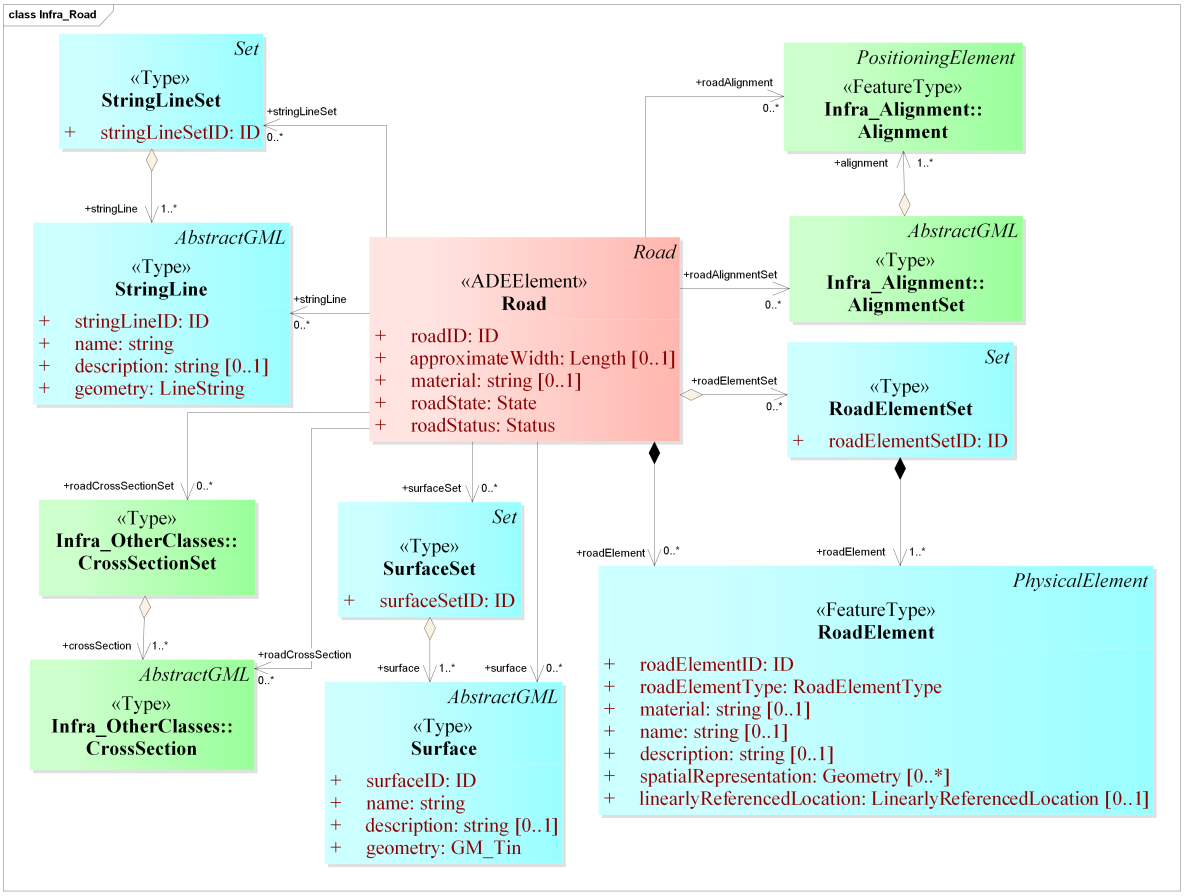


Figure 3. The UML excerpt depicts existing CityGML class `CG::Road` (marked with stereotype «ADEElement») extended to include attributes and classes from LandInfra `LI::Road` class in the CityGML Infra ADE.

`CG::TINRelief` is extended to include `LI::LandSurface`'s attributes. It stores:

- The ID and the scope of the land surface features present in the dataset (`<infra:landSurfaceID>`);
- The material of the land surface (`<infra:material>`);
- The spatial representation of the land surface as TINs (similar to `TINRelief`);
- The attribute indicating if the feature is existing or proposed (`<infra:state>`).

A LandInfra `LI::LandElement` feature can be a terrain, water body or vegetation depending upon the value of its `elementType`. Therefore, the CityGML classes `CG::WaterBody`, and *`CG::_VegetationObject`* (`CG::PlantCover` and `CG::SolitaryVegetationObject`) are extended to include `LandElement`'s attributes, such as ID (`<infra:landElementID>`), type of land element (`<infra:landElementType>`), material (`<infra:material>`), state (`<infra:state>`), property (`<infra:property>`) and sets of associated properties (`<infra:propertySet>`), and so on.

Similarly, CityGML `CG::Building` and `CG::BuildingPart` are extended to include LandInfra `LI::Building`'s (from `LI::FacilityPart`) attributes such has ID, state and status. See Appendix A for the rest of the UML models.

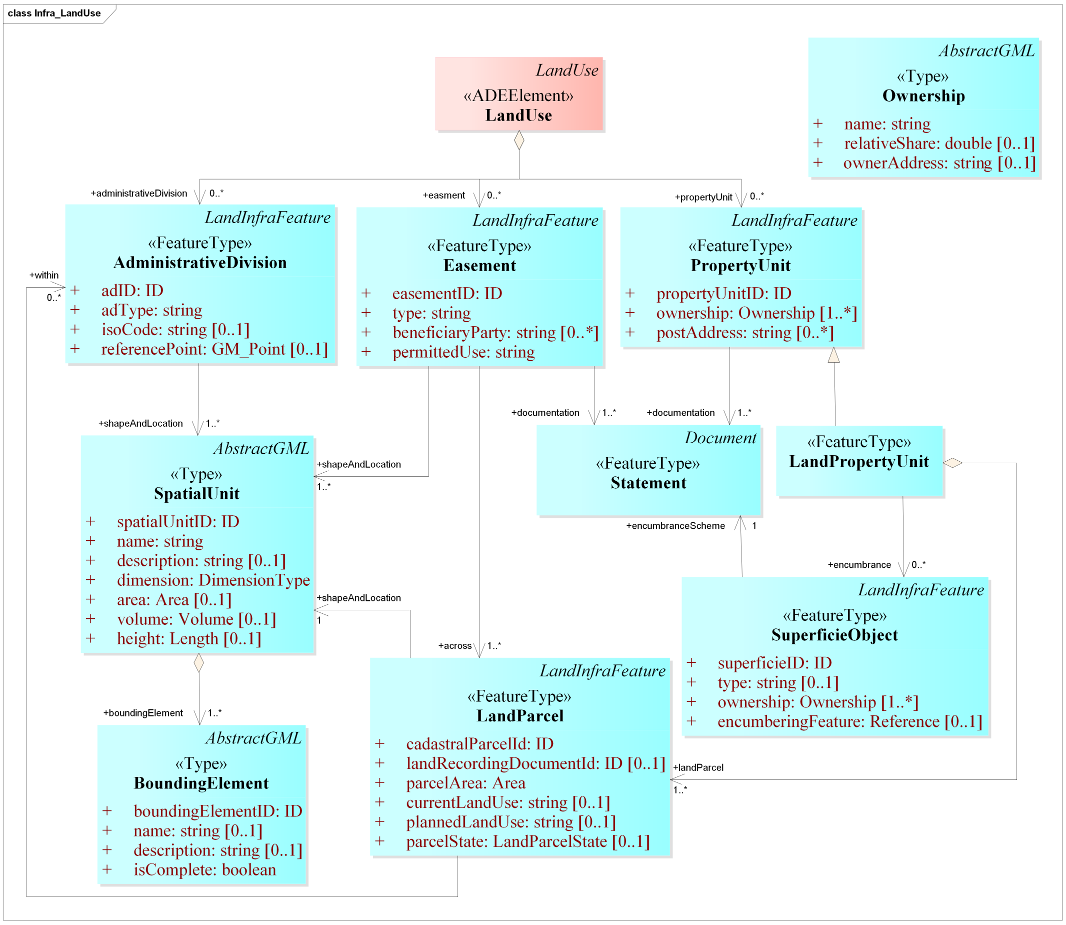


Figure 4. The UML excerpt depicts existing CityGML `CG::LandUse` (marked with stereotype «ADEElement») extended to include attributes and classes from LandInfra `LI::LandDivision`, in the CityGML Infra ADE.

4.1.2. Adding New Feature Types for Non-Matching LandInfra Classes

A new feature type `LandInfraFeature` is introduced as a subclass of *`CG::_CityObject`* to represent LandInfra's `LI::Feature` class. Since `LI::Feature` is a concrete class and *`CG::_CityObject`* is an abstract class, we introduced `LandInfraFeature` as a concrete class in our ADE (Figure 1).

LandInfra `LI::Document` is introduced to store documents with information about the datasets, e.g., statements, condominium schemes, etc. (Figure 1). Similarly, LandInfra's `LI::SurveyMark` is defined to store points on the surface of the Earth which are stable during surveying operations.

LandInfra `LI::Facility` and `LI::FacilityPart` are introduced as the subclasses of `LandInfraFeature` to represent the infrastructure facilities in the Infra ADE. Further, to store the activities related to the improvement of facilities, such as design and/or construction, we introduced the LandInfra `LI::Project` and `LI::ProjectyPart` feature types in the ADE (Figure 1).

LandInfra's `LI::Alignment` and its associated classes are introduced in the ADE to provide a linear referencing system for locating the features, e.g., an alignment for the centreline of a road, alignment for rails, etc. (Figure 5). An alignment can be represented as:

- A simple 2D line string (`<infra:lineString2DRepresentation>`);
- A horizontal alignment (`<infra:Alignment2DHorizontal>`);
- A horizontal alignment with an accompanying 2D vertical long section taken along the horizontal alignment (`<infra:Alignment2DVertical>`);
- A 3D line string (`<infra:lineString3DRepresentation>`).

LandInfra `LI::CondominiumBuilding`, `LI::CondominiumBuildingPart` and other associated classes are introduced in the Infra ADE. The LandInfra abstract class *`LI::_LandLayer`* is introduced as it is to represent the layers underneath the land surface (Figure 6). They can be defined in three ways: As a 3D polyface mesh solid (`<infra:SolidLayer>`), as a collection of surface layers (`<infra:SurfaceLayer>`), or as a series of 2D vertical cross sections (`<infra:LinearLayer>`). Lastly, LandInfra `LI::Survey` is introduced in the ADE to model information related to the acquisition of geometry and semantic properties of features.

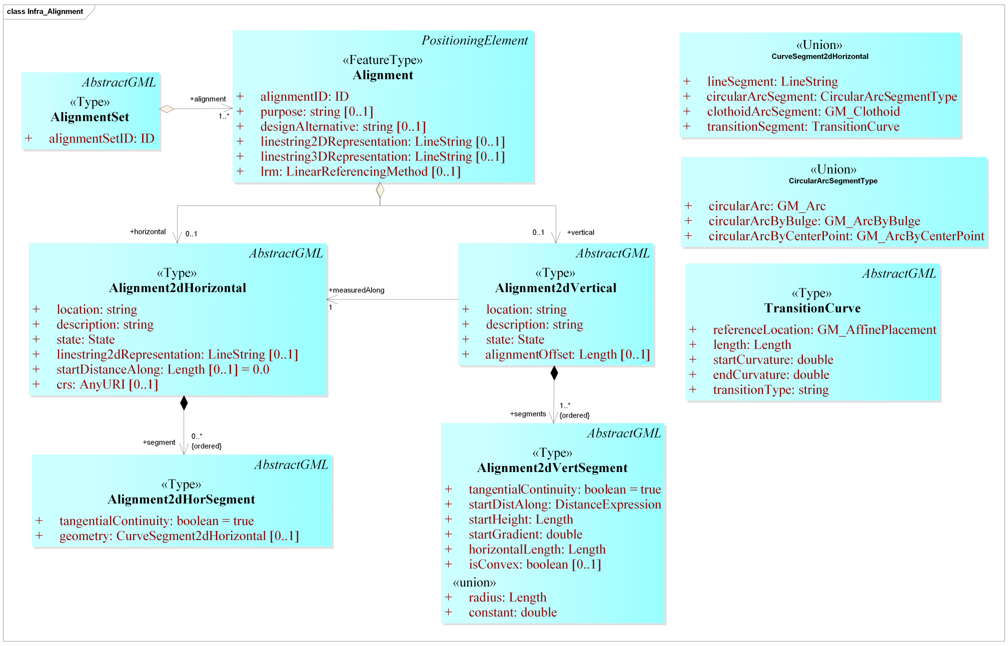


Figure 5. The UML excerpt depicts the `Infra_Alignment` introduced in the Infra ADE for linear referencing of the features.

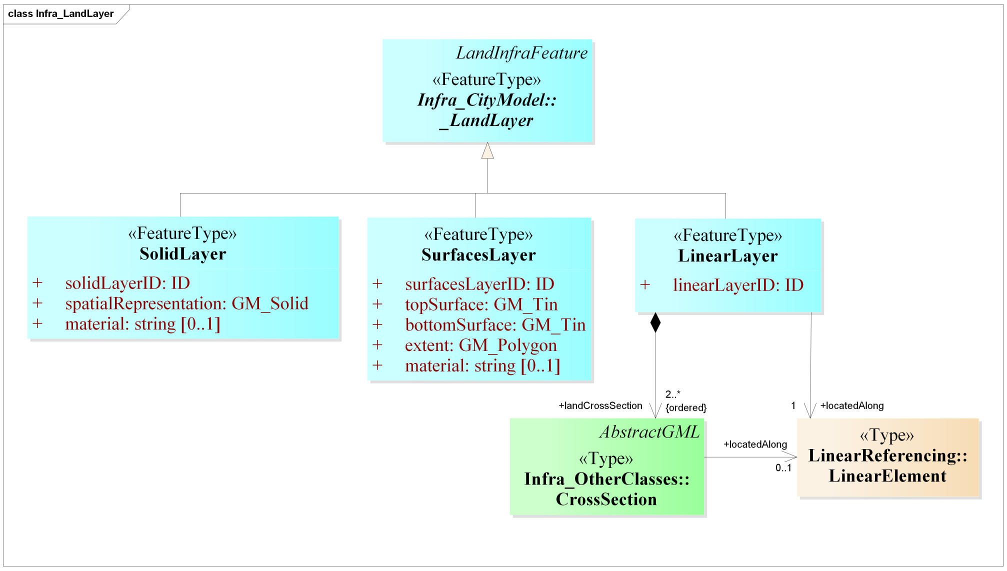


Figure 6. The UML excerpt depicts new LandInfra feature *`_LandLayer`* introduced in the CityGML Infra ADE to represent the layers underneath the land surface.

4.1.3. New Geometry Types, Data Types and Codelists

Three new LandInfra-specific geometry types are introduced in the CityGML Infra ADE, namely `IndexedPoint`, `SimpleIndexedPolygon`, and `PolyfaceMesh`. These geometry types do not exist in the ISO 19107 and in GML. Given that existing software (FME [31], FZK viewer [32], etc.) would require additional implementation to support and visualize these new geometry types, we made their implementation optional in the ADE (in case there is support in the future). We fully understand how difficult it can be to extend software support for each and every ADE that defines new geometry types. Therefore, while converting from InfraGML to CityGML Infra ADE using our software prototype, these geometry types are now converted to the existing OGC Simple Feature structure supported in GML.

Further, one new LandInfra data type, `ID` is introduced. The `ID` data type is defined to uniquely identify the features within the scope of the dataset. It has an attribute `identifier` which is a user defined ID unique within the dataset or globally unique with the inclusion of `scope` attribute (see snippet below).

```
<li:ID>
<li:identifier>GML_e8e7963f-718c-40fb-8253-753f2d468f0f</li:identifier>
<li:scope>OGC LandInfraSWG</li:scope>
</li:ID>
```

We also defined 18 new codelists that were taken from LandInfra, which are summarised in Table 3. They are implemented as simple dictionaries according to the CityGML specifications and can be further extended. It is interesting to note that the codelist used to identify the type of easement (`EasementType`) in the LandInfra requirement class `LandDivision` is missing in the LandInfra specifications. It is therefore not included in the codelists defined for the ADE. We also defined two enumerations taken from LandInfra: `Side` and `StringDirectionType`.

Table 3. Description of codelists included in the Infra ADE.

#	**Codelist**	**Description**
1	State	Whether an object is existing or proposed.
2	Status	Life cycle stage of an object, e.g., planned, under construction, etc.
3	ProfessionalType	Position of the person in charge of the land development and infrastructure project e.g., draftsman, engineer, surveyor, etc.
4	FacilityPartType	e.g., building, road, etc.
5	LandElementType	e.g., vegetation, terrain, etc.
6	RoadElementType	e.g., pavement, gutter, etc.
7	RailwayElementType	e.g., switch, rail, etc.
8	LandParcelState	e.g., main parcel or carrier parcel, etc.
9	SurveyMonumentType	e.g., boundary marker, observation point of geodetic significance, etc.
10	StatementType	Statement that is signed to establish the interest in the land.
11	SigningRole	Role a signing party plays, e.g., owner, buyer, seller, etc.
12	CondominiumUseType	e.g., residential, office, etc.
13	BuildingPartType	e.g., the main part to which the postal address refers to, or a secondary part like the basement of a shop, etc.
14	StringType	Geometric string representation of bounding element `BEString`.
15	DimensionType	Dimension of a spatial unit. Depending on the value, a spatial unit can include attributes such as area, volume and height.
16	ImplicitSurfaceType	Top or bottom surface of a spatial unit for the bounding element `BEImplicitFace`.
17	SurveyType	e.g., compiled, computed or actually surveyed, etc.
18	SurveyResultType	e.g., point, point cloud, image, etc.

5. Implementation and Testing

5.1. Software Prototypes

In order to test our ADE and show its usability, we have developed an open source prototype that automatically converts datasets from InfraGML to CityGML (with our Infra ADE), and vice versa. For the conversion from CityGML to InfraGML, since InfraGML does not offer the possibility to extend its core model, only the classes and attributes that can be mapped are converted.

It should be noticed that we do not attempt to convert to a harmonised data model (see El-Mekawy et al. [33] as an example) since the resulting files would not be useful in practice: Software packages do not have support for such data models.

The software we have developed, together with sample datasets, is freely available in our GitHub repository: https://github.com/tudelft3d/city2InfraGML. It is composed of two Python scripts:

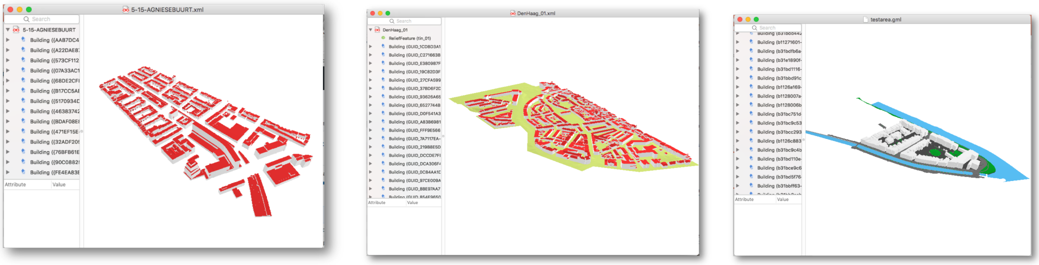
1. *citygml2infragml.py* for converting original CityGML models to InfraGML.
2. *infragml2citygml.py* for converting InfraGML models to CityGML (with Infra ADE).

In addition, since there is currently no official way to validate InfraGML datasets, we developed a validator that checks a dataset against the schema (https://github.com/tudelft3d/city2InfraGML). It can be combined with val3dity (https://github.com/tudelft3d/val3dity) to validate the geometry of the 3D primitives according to the international standard ISO 19107 [34,35]. For the validator, we introduced an additional wrapper schema for specifically validating different LandInfra features (e.g., terrain, facilities, roads, etc.) within a single dataset.

5.2. Experiments and Validation

We tested our software with various real world datasets in the Netherlands (Figure 7) and validated our results by checking them against the schema using the validator. The examples of XML in the following sections are taken from the real-world datasets that we have created.

The tested datasets were the following.



(**a**) Rotterdam (**b**) Den Haag (**c**) Delft

Figure 7. CityGML datasets used for testing the prototypes, visualised in azul.

1. From CityGML to InfraGML:
   (a) CityGML 2.0.0 LOD2 (Levels of Detail) city model of an area in Rotterdam, the Netherlands (Source: http://rotterdamopendata.nl/dataset/rotterdam-3d-bestanden).
   (b) CityGML 1.0.0 LOD2 models of building and terrain of an area in the Hague, the Netherlands (Source: https://data.overheid.nl/data/dataset/48265-3d-lod2-stadsmodel-2010-den-haag-citygml).
   (c) CityGML 2.0.0 LOD1 city model of an area of Delft, the Netherlands generated using 3dfier (https://github.com/tudelft3d/3dfier) (Source: https://3d.bk.tudelft.nl/opendata/3dfier/).

2. From InfraGML to CityGML:
   (a) Sample InfraGML 1.0.0 datasets of land surfaces, facilities and roads.
   (b) 3D city models of an area in Delft, Rotterdam, and the Hague, the Netherlands generated by converting original CityGML LOD1 and LOD2 models to InfraGML using *citygml2infragml.py*.

6. Use Cases for the CityGML Infra ADE

Some use cases for LandInfra are included in the official documentation [1], which would be equally applicable to the Infra ADE. These are: Road alignments, surveying, conversions between LandXML and InfraGML, storage of terrain data, land division, and representation of railway features. In addition, Blanchet et al. [27] investigated whether CityGML or InfraGML is best suited for initial environment acoustic studies, but the research was limited to a conceptual study of the LandInfra standard, and real world InfraGML datasets were not available.

We provide here a list of additional use cases where our CityGML Infra ADE can be useful in practice:

1. Subsurface modelling

   CityGML originally does not model real-world subsurface data originating from the geological models [36], which is useful for many applications, such as infrastructural works that require excavations and soil studies. Since LandInfra has support for modelling topography (terrain) and subsurface information in its requirement class `LandFeature`, our Infra ADE enables the modelling of surface features (such as buildings, roads, etc.) with subsurface information in an integrated framework (see Snippet 1 below for implementation in an Infra ADE dataset). The subsurface layers can be represented in three ways in the Infra ADE: As TINs, 3D polyface mesh solids, or vertical 2D cross sections. Each subsurface layer can have an additional attribute `material` to specify the material of the layer. This integrated framework will not only benefit the planning and design process for surface and subsurface structure construction, but also make transparent the risk management.

Snippet 1: CityGML Infra ADE with an LOD1 Building and a subsurface layer.

```xml
<CityModel>
<!-- CityGML LOD1 Building with Infra ADE attributes -->
<cityObjectMember>
<bldg:Building gml:id="building01">
<!-- CityGML attributes -->
<gml:name>CityGML Infra ADE LOD1 Building Model</gml:name>
<creationDate>2016-11-24</creationDate>
<bldg:function>1004</bldg:function>
<bldg:measuredHeight uom="m">4.123</bldg:measuredHeight>
<bldg:lod1Solid/>
<bldg:address/>
<!-- Infra ADE attributes -->
<infra:buildingID>
<infra:ID>
<infra:identifier>building01</infra:identifier>
</infra:ID>
</infra:buildingID>
<infra:buildingState>existing</infra:buildingState>
<infra:buildingStatus>constructed</infra:buildingStatus>
</bldg:Building>
```

```
</cityObjectMember>
<!-- CityGML Infra ADE subsurface layer (SolidLayer) -->
<cityObjectMember>
<infra:SolidLayer gml:id="layer01">
<infra:state>existing</infra:state>
<infra:spatialRepresentation xlink:href="#pmesh1"/>
<infra:solidLayerID>
<infra:ID>
<infra:identifier>laye01</infra:identifier>
</infra:ID>
</infra:solidLayerID>
<infra:material>clay</infra:material>
</infra:SolidLayer>
</cityObjectMember>
</CityModel>
```

2. 3D cadastre

CityGML lacks the capability to represent the legal extents and rights of entities within a complex, which is the core of land administration information. Extensions have been proposed to integrate this information in the data model of CityGML for the management of property rights, e.g., the CityGML Land Administration Domain Model (LADM) ADE to represent the legal ownership of buildings and their parts in CityGML in accordance with the ISO 19152-LADM [37] standard [38,39]. However, most of the available land administration research with CityGML is centred around buildings.

LandInfra is more than LADM. LandInfra addresses land development in the context of activities concerning civil engineering infrastructure facilities [1]. This is achieved by modelling what is needed to account for such activities, including defining the legal entities, their boundaries, as well as identification of the signing parties [1]. LandDivision is one of the requirement classes of LandInfra. As mentioned in the LandInfra specifications, the scope of LandInfra does not include land recording and database storage. The LandInfra Standard addresses only a subset of LADM [1]. The integration of LandInfra with CityGML in our Infra ADE further enables modelling administrative divisions, cadastral information and ownership rights of condominiums, and subsurface infrastructure such as underground tunnels (Snippet 2).

Snippet 2: CityGML LandUse extended with administrative divisions in Infra ADE.

```
<CityModel>
<cityObjectMember>
<!-- CityGML LandUse -->
<luse:LandUse gml:id="luse01">
<!-- Infra AdministrativeDivision -->
<infra:administrativeDivision>
<infra:AdministrativeDivision gml:id="admin01">
<infra:state>existing</infra:state>
<infra:adID>
<infra:ID>
<infra:identifier>admin01</infra:identifier>
</infra:ID>
</infra:adID>
<infra:adType>Municipality</infra:adType>
<infra:shapeAndLocation>
<infra:SpatialUnit gml:id="su01">
```

```
<infra:spatialUnitID>
<infra:ID>
<infra:identifier>su01</infra:identifier>
</infra:ID>
</infra:spatialUnitID>
<infra:dimension>3D</infra:dimension>
<infra:boundingElement>
.... <!-- Any of the Bounding elements -->
</infra:boundingElement>
</infra:SpatialUnit>
</infra:shapeAndLocation>
</infra:AdministrativeDivision>
</infra:administrativeDivision>
</luse:LandUse>
</cityObjectMember>
</CityModel>
```

3. Urban facility management

Currently, most of the research related to facility management is confined to buildings. For instance, Kim et al. [40] implemented the CityGML Indoor ADE to implement indoor space and indoor facility management applications for buildings. Similarly, the CityGML Computer Aided Facility Management (CAFM) ADE was developed by Moshrefzadeh et al. [41] to integrate detailed geometric and semantic information on the outer shell of the buildings for applications like cleaning management, and cost planning and management.

In LandInfra, a facility includes buildings and other infrastructure, such as roads, railways, runways, waste water system, bridge, utilities (pipelines) etc. [1]. The integration of LandInfra with CityGML in our Infra ADE enables effective management of all the aforementioned facilities in CityGML. Each facility, whether a building or a road, has a life cycle, including planning, design, construction, maintenance, operation, and termination phases. Furthermore, a facility may be broken down into parts (`FacilityPart`), e.g., a shopping mall may include buildings, roads, site, drainage, water distribution and waste water [1]. Any activity such as the design or construction related to a facility (or its parts) is managed through projects (`Project/ProjectPart`). The CityGML Infra ADE dataset can include any number of projects to store the status of the facility project (`projectStatus`) and the date on which the status value is valid (`statusDate`) to make the dataset more manageable (Snippet 3).

Snippet 3: CityGML Infra ADE with project data for facility management.

```
<CityModel>
<!-- CityGML Infra ADE facility with two parts: Building & Road-->
<cityObjectMember>
<infra:Facility gml:id="facility01">
<infra:state>existing</infra:state>
<infra:facilityID>
<infra:ID>
<infra:identifier>facility1</infra:identifier>
</infra:ID>
</infra:facilityID>
<infra:type>shopping mall</infra:type>
<infra:status>under constructed</infra:status>
<infra:part xlink:href="#Building01"/>
<infra:part xlink:href="#Road01"/>
```

```
</infra:Facility>
</cityObjectMember>
<!-- CityGML Building -->
<cityObjectMember>
<bldg:Building gml:id="Building01">
....
</bldg:Building>
</cityObjectMember>
<!-- CityGML Road -->
<cityObjectMember>
<tran:Road gml:id="Road01">
....
</tran:Road>
</cityObjectMember>
<!-- CityGML Infra ADE Project for the whole facility-->
<cityObjectMember>
<infra:Project gml:id="Project01">
<infra:state>existing</infra:state>
<infra:projectID>
<infra:ID>
<infra:identifier>Project01</infra:identifier>
</infra:ID>
</infra:projectID>
<infra:projectStatus>under construction</infra:projectStatus>
<infra:statusDate>2019-01-01</infra:statusDate>
<!-- CityGML Infra ADE ProjectPart for the constructed FacilityPart Building-->
<infra:projectPart>
<infra:ProjectPart gml:id="ProjectPart1">
<infra:state>existing</infra:state>
<infra:projectPartID>
<infra:ID>
<infra:identifier>ProjectPart1</infra:identifier>
</infra:ID>
</infra:projectPartID>
<infra:status>constructed</infra:status>
<infra:statusDate>2019-01-01</infra:statusDate>
<infra:facilityPart xlink:href="Building01"/>
</infra:ProjectPart>
</infra:projectPart>
<!-- CityGML Infra ADE ProjectPart for the under construction FacilityPart Road-->
<infra:projectPart>
<infra:ProjectPart gml:id="ProjectPart2">
<infra:state>existing</infra:state>
<infra:projectPartID>
<infra:ID>
<infra:identifier>ProjectPart2</infra:identifier>
</infra:ID>
</infra:projectPartID>
<infra:status>under construction</infra:status>
<infra:statusDate>2019-01-01</infra:statusDate>
<infra:facilityPart xlink:href="Road01"/>
</infra:ProjectPart>
</infra:projectPart>
</tran:Road>
</cityObjectMember>
```

```
</CityModel>
```

4. Surveying

Survey data can be used as a reliable data source at all the stages of the life cycle of a building or other features. Designers of architectural and design projects, armed with accurate site data, can work with reduced overall commercial risk, and with greater certainty. CityGML does not provide a way to store survey data in a structured way, but LandInfra has a requirement class `Survey` outlining the specifications to store such data. Because of this, the integration of LandInfra with CityGML in our Infra ADE enables effective management of survey metadata in CityGML, e.g., survey type and its purpose, surveyor information, and so on. Further, it is possible to store information about the survey observation points, accuracy information, equipments or sensors used, and the results.

Snippet 4: CityGML Infra ADE with Survey data

```
<CityModel>
<infra:survey>
<infra:Survey gml:id="Survey01">
<infra:surveyID>
<infra:ID>
<infra:identifier>Survey01</infra:identifier>
</infra:ID>
</infra:surveyID>
<infra:name>SurveyData</infra:name>
<infra:description>Dataset with Survey data</infra:description>
<infra:type>computed</infra:type>
<!-- Data about the field notes taken during the survey -->
<infra:fieldNote>
<infra:Document gml:id="SurveyDoc01">
<infra:state>existing</infra:state>
<infra:documentID>
<infra:ID>
<infra:identifier>SurveyDoc01</infra:identifier>
</infra:ID>
</infra:documentID>
<infra:documentType>Open Data</infra:documentType>
<infra:documentContent>www.abc.com/abc.pdf</infra:documentContent>
</infra:Document>
</infra:fieldNote>
<infra:setup>
.... <!-- Survey setup data -->
</infra:setup>
<infra:equipment>
.... <!-- Equipments used for surveying -->
</infra:equipment>
<infra:surveyResult>
.... <!-- Results of the survey here -->
</infra:surveyResult>
</infra:Survey>
</infra:survey>
</CityModel>
```

5. Asset management

   Asset management can be summarised as a systematic approach to the process of maintaining, upgrading and operating physical assets in a cost-effective way for both short- and long-term planning [42]. For municipalities and regional governments asset management is a crucial element of day-to-day operations. Within asset management, the maintenance of roads and transportation networks is key to keeping traffic moving smoothly and safely on a daily basis. Road maintenance includes activities such as smoothness control/de-icing, repairs, closures, milestone maintenance and traffic city furniture replacement [17].

   While CityGML has support for Levels of Detail (LODs), it only supports line representation at LOD0 and polygon representation at LOD1–LOD4. InfraGML supports four representations for roads: Solid, faceted (triangular) surfaces, lines running longitudinally and 2D views cut perpendicular to a road's centreline. It is also possible to model the cross-section of a road, which is valuable for repair projects. Furthermore, CityGML has support for railways in its `Transportation` module, but it has little support and almost no documentation. It is also unclear how to exactly model a railway in CityGML, as it is also mentioned as being a part of the `Tunnel` and `Bridge` modules. With InfraGML, there is a dedicated class for railways and this has support for 3D railway elements and track geometry including superelevation (cant). Snippet 5 summarises the potential additional elements from LandInfra that can enhance CityGML data for road asset management.

Snippet 5: CityGML Infra ADE with project data for road asset management.

```
<CityModel>
<cityObjectMember>
<tran:Road gml:id="road_01">
<gml:name>Main Street</gml:name>
<creationDate>2016-11-24</core:creationDate>
<tran:class>350000</tran:class>
<tran:function>1</tran:function>
<tran:usage>2</tran:usage>
<tran:lod2MultiSurface/> ### Geometry Here ###
<infra:roadID>
<infra:ID>
<infra:identifier>road_01</infra:identifier>
</infra:ID>
</infra:roadID>
<infra:approximateWidth uom="m">5</infra:approximateWidth>
<infra:roadElement xlink:href="#re1"></infra:roadElement>
<infra:roadAlignment xlink:href="#Alignment1"></infra:roadAlignment>
<infra:roadElement>
<infra:RoadElement>
<gml:description>6.5 cm asphalt top surface course</gml:description>
<gml:name>top pavement layer</gml:name>
<infra:state>existing</infra:state>
<infra:spatialRepresentation/> <!-- Spatial Representation Here -->
<infra:roadElementID>
<infra:ID>
<infra:identifier>pavement1</infra:identifier>
</infra:ID>
</infra:roadElementID>
<infra:roadElementType>Pavement Surface Course</infra:roadElementType>
<infra:material>asphalt</infra:material>
```

```
</infra:RoadElement>
</infra:roadElement>
</tran:Road>
</cityObjectMember>
</CityModel>
```

6. *Urban environmental analysis*

   The added value of integrating CityGML and LandInfra in our Infra ADE can also be seen in urban applications such as estimating the level of noise exposure on buildings, or how much solar irradiation a building will receive. Unlike CityGML, LandInfra explicitly models the materials of road surfaces and terrain, geometry and semantics of railways, type of road elements (pavements, hard shoulders, soft shoulders etc.), construction materials of buildings, and information about the observation/measurement points to name a few. Such information is useful for environmental applications such as urban noise and flood mapping. We have added all these elements in the Infra ADE to supplement environmental analysis using CityGML. The previous snippet of XML Snippet 5 contains elements that can be useful in urban environmental analysis.

7. Conclusions and Future Work

LandInfra is a more powerful standard than CityGML in some areas, as it has a much more detailed representation for land and infrastructure features. However, it currently has essentially no support in software, and even the academic papers that touch upon its theoretical potential do not use it in practice. Our Infra ADE for CityGML provides a way to change this situation by embedding LandInfra's features in CityGML. This way, we can use the best of both standards, and we can also ensure that the resulting datasets can be used in practice by the software packages already supporting CityGML. There is a lot of support available for the CityGML extension mechanism such as parsers, validators, DBMS, and so on, which can indirectly contribute to the adoption of LandInfra through our ADE. One example is the latest version of *3DCityDB* (3D City Database 4.0.0), which offers support to store CityGML files having ADEs in a database. A CityGML ADE is handled like a 'plugin' and the 3DCityDB core database schema is extended dynamically with new tables based on the schemas of the ADE [43,44]. Similarly, *citygml4j* [45], a Java API for CityGML, supports reading and writing CityGML ADE datasets. Further, it is also possible to visualize the ADE datasets with new city objects (with GML geometries) and semantic attributes using FME and FZK viewer. As discussed before, these software would require additional implementation to support and visualize any new geometry types (other than GML) introduced in an ADE dataset.

In order to develop the Infra ADE, we have performed a detailed analysis of the individual classes, attributes and relations in CityGML and LandInfra, and created a mapping from LandInfra to CityGML. We have mapped LandInfra to CityGML (and not vice versa) because CityGML provides mechanisms to extend its data model with new feature types and attributes using Generic city objects or ADEs, whereas similar extensions are not supported in the LandInfra standard. Moreover, CityGML has the concept of LODs, which is widely used in practice and missing in LandInfra.

To provide a proof-of-concept of our mapping, we have developed two open source software prototypes for converting CityGML (and Infra ADE) datasets to InfraGML and vice versa. Since LandInfra is a relatively new standard and there are no concrete datasets available for it, the developed prototypes can help practitioners to generate valid real-world sample InfraGML datasets, which can then lead to the real-world applications that are currently missing from the standard.

Furthermore, we are working on the development of InfraJSON, a JSON (JavaScript Object Notation) based encoding for InfraGML, which can be explored at our public GitHub repository: https://github.com/tudelft3d/InfraJSON. Despite the precedence set by the high usage of GML (XML) in various OGC standards, it is a verbose and complex encoding for use in real world applications, whereas JSON provides an easy-to-use and easy-to-read alternative [46,47]. For instance, CityJSON (http://www.cityjson.org) is a JSON-based encoding for a subset of the OGC CityGML data model. Virtually all features are supported except a few (https://www.cityjson.org/citygml-compatibility/). It also has support for software tools available for generation, manipulation and visualisation. InfraJSON is a work in progress and its development would encourage the usage of the LandInfra standard.

Author Contributions: K.K. conceived the paper, and carried out the literature review together with A.L. and K.A.O., K.K. and A.L. developed the methodology. K.K. performed the software implementation. K.K., A.L., and K.A.O. wrote the paper in consultation with H.L. and J.S. All authors read and approved the final manuscript.

Funding: The research leading to this paper is a part of the research project 3D4EM (3D for Environmental Modelling) in the Maps4Society programme (grant No. 13740) which is funded by the NWO (Netherlands Organisation for Scientific Research), and partly funded by the Ministry of Economic Affairs. This work is also funded by the European Research Council under the European Union's Horizon 2020 ERC Agreement no. 677312 UMnD: Urban modelling in higher dimensions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADE	Application Domain Extension
BIM	Building Information Modelling
FME	Feature Manipulation Engine
GIS	Geographical Information Systems
GML	Geography Markup Language
IFC	Industry Foundation Classes
ISO	International Standards Organization
JSON	JavaScript Object Notation
LOD	Level Of Detail
OGC	Open Geospatial Consortium
TIN	Triangulated Irregular Network
UML	Unified Modelling Language
XML	eXtensible Markup Language

Appendix A

The remaining UML models of the classes and attributes introduced in the CityGML Infra ADE are presented in this section.

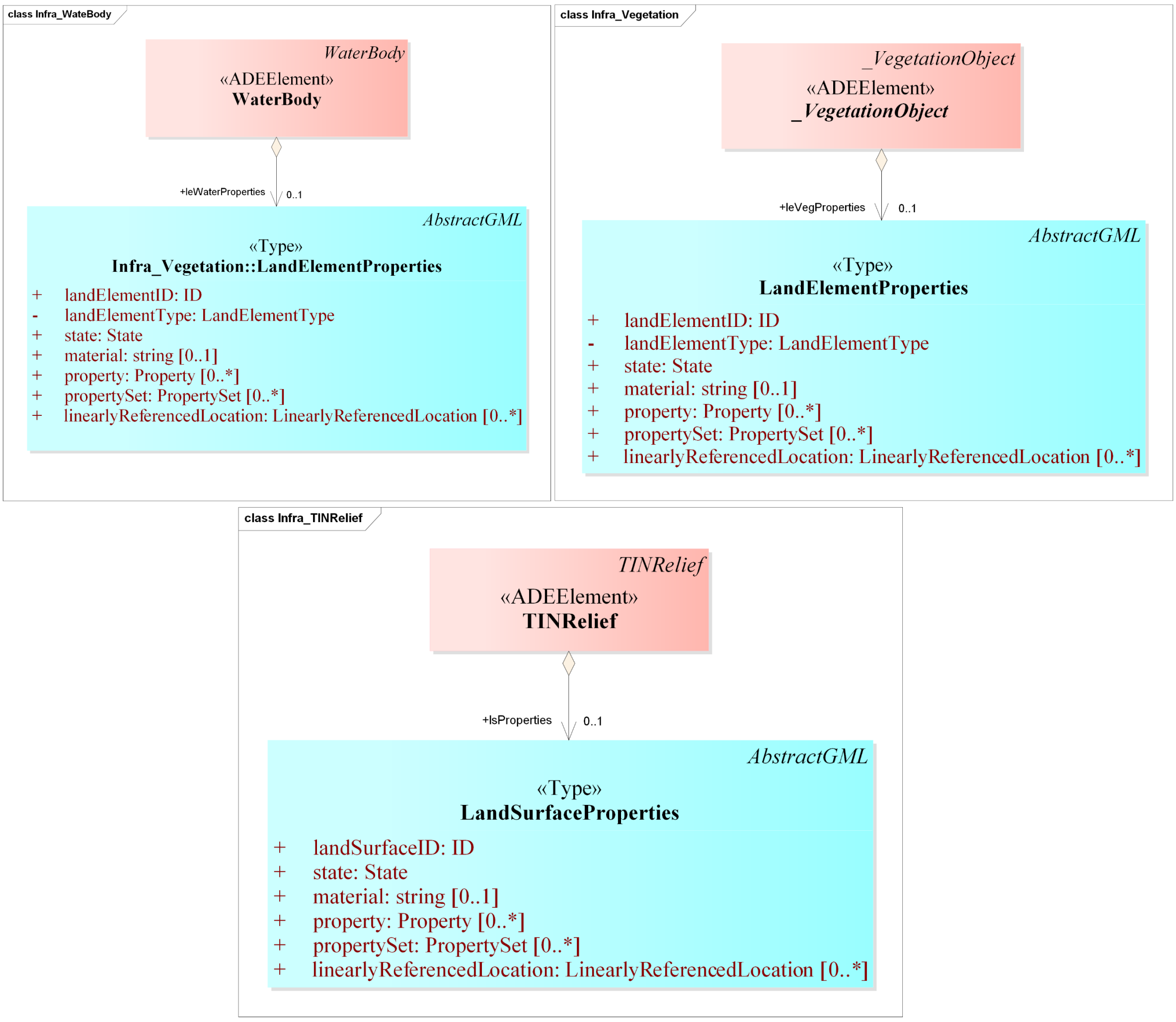

class Infra_WateBody
WaterBody
«ADEElement»
WaterBody
+leWaterProperties
0..1
AbstractGML
«Type»
Infra_Vegetation::LandElementProperties
+ landElementID: ID
- landElementType: LandElementType
+ state: State
+ material: string [0..1]
+ property: Property [0..*]
+ propertySet: PropertySet [0..*]
+ linearlyReferencedLocation: LinearlyReferencedLocation [0..*]
class Infra_Vegetation
_VegetationObject
«ADEElement»
_VegetationObject
+leVegProperties
0..1
AbstractGML
«Type»
LandElementProperties
+ landElementID: ID
- landElementType: LandElementType
+ state: State
+ material: string [0..1]
+ property: Property [0..*]
+ propertySet: PropertySet [0..*]
+ linearlyReferencedLocation: LinearlyReferencedLocation [0..*]
class Infra_TINRelief
TINRelief
«ADEElement»
TINRelief
+lsProperties
0..1
AbstractGML
«Type»
LandSurfaceProperties
+ landSurfaceID: ID
+ state: State
+ material: string [0..1]
+ property: Property [0..*]
+ propertySet: PropertySet [0..*]
+ linearlyReferencedLocation: LinearlyReferencedLocation [0..*]

Figure A1. The UML excerpt depicts existing CityGML classes of `CG::TINRelief`, `CG::WaterBody`, `CG::PlantCover`, `CG::SolitaryVegetationObject` extended to include attributes from LandInfra `LI::LandSurface`, and `LI::LandElement` classes, respectively, in the CityGML Infra ADE.

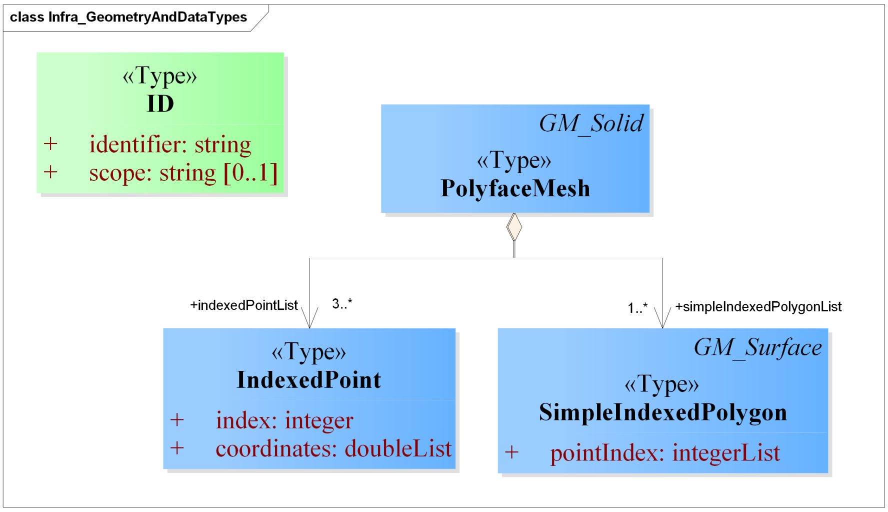


Figure A2. The UML excerpt depicts the new data type `ID` and three new LandInfra specific geometry types implemented in the CityGML Infra ADE.

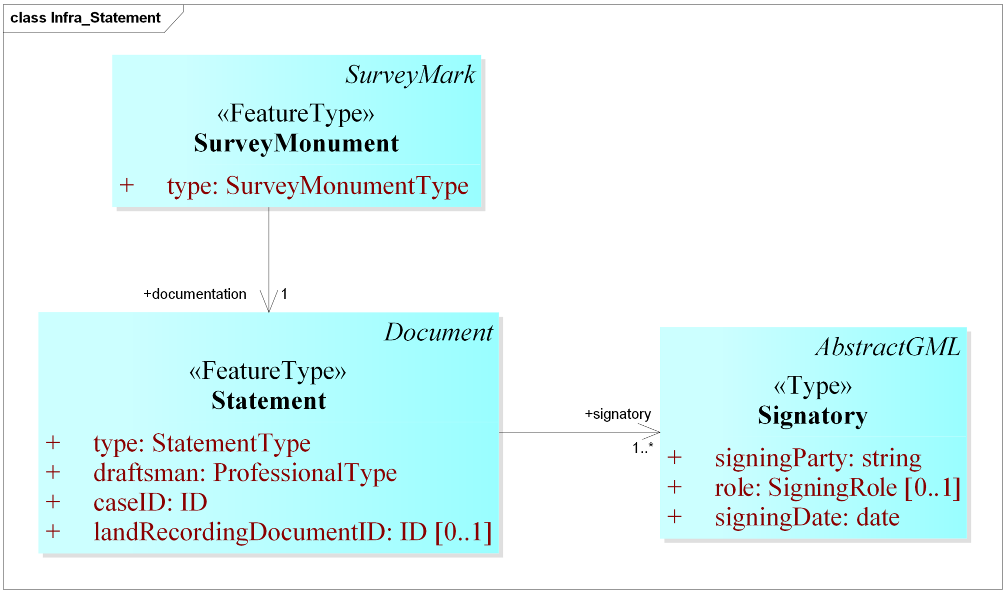


Figure A3. The UML excerpt depicts the LandInfra classes `SurveyMonument` and `Statement` in the CityGML Infra ADE.

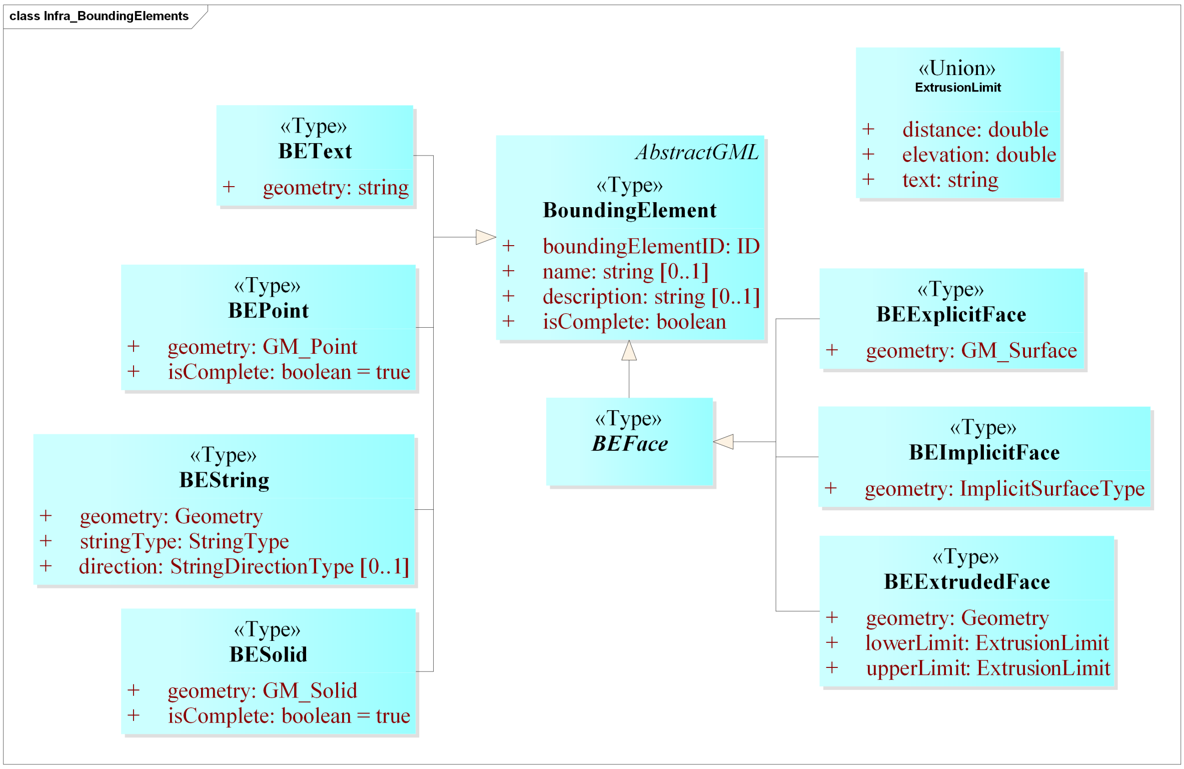


Figure A4. The UML excerpt depicts the LandInfra `BoundingElement(s)` to specify the boundary of the spatial units in the CityGML Infra ADE.

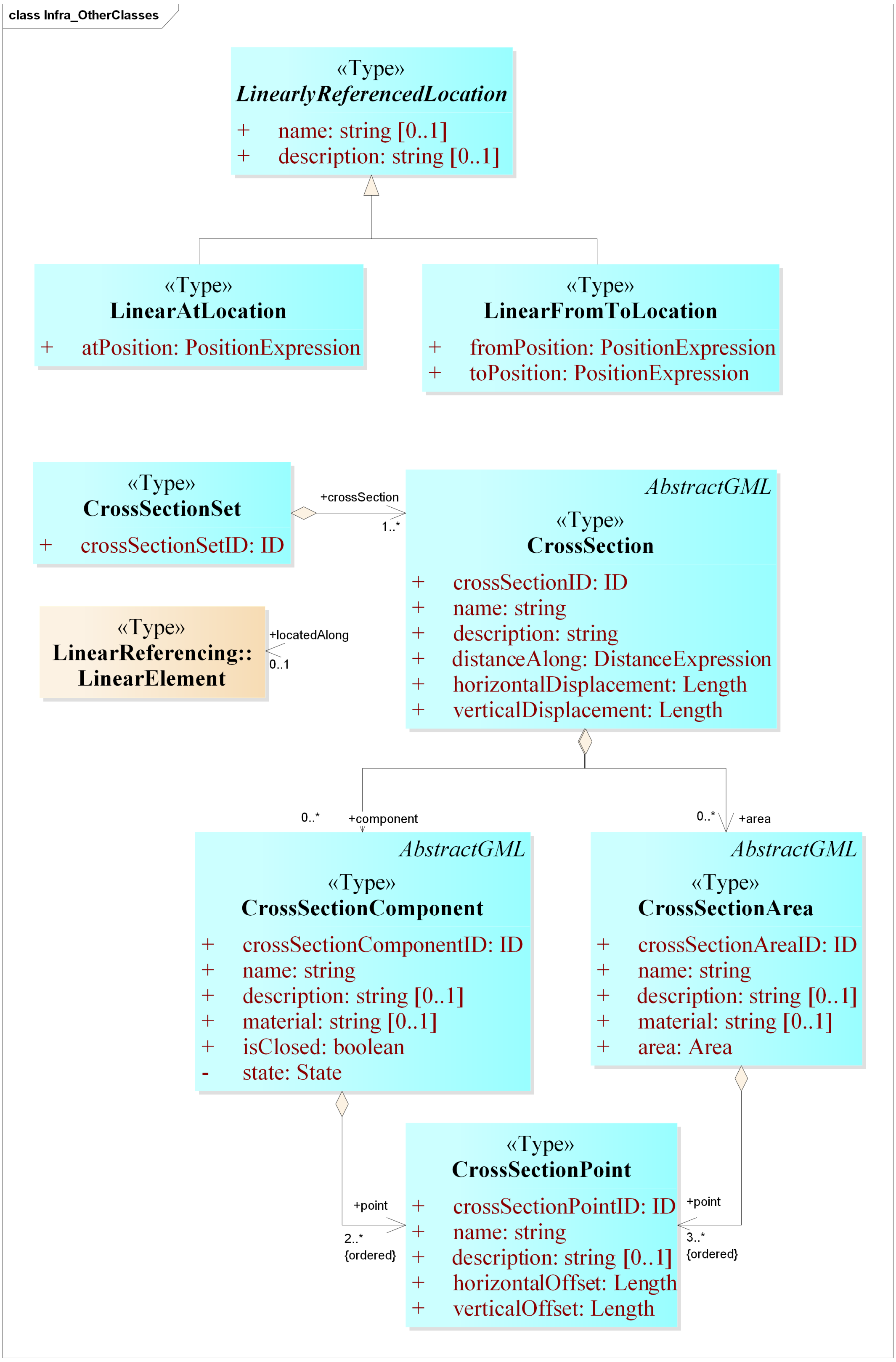


Figure A5. The UML excerpt depicts the LandInfra classes `CrossSection` and *`LinearlyReferencedLocation`* in the CityGML Infra ADE.

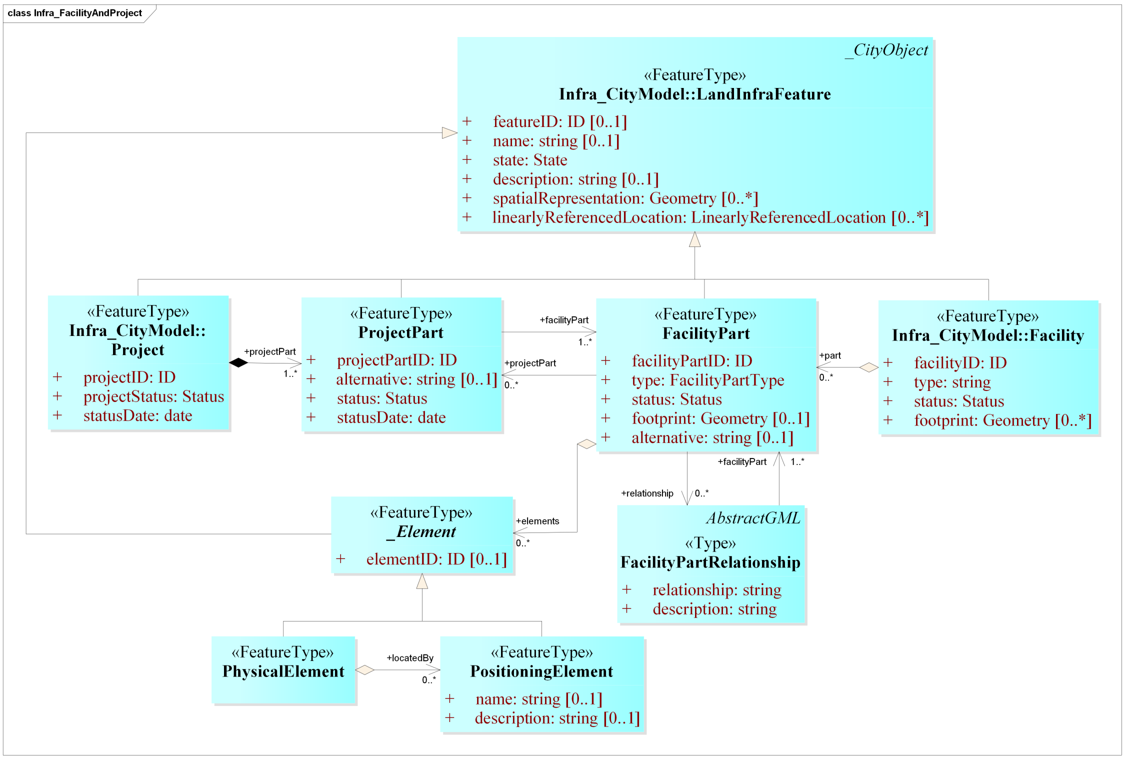


Figure A6. The UML excerpt depicts new LandInfra features `Facility`, and `Project`) introduced in the CityGML Infra ADE.

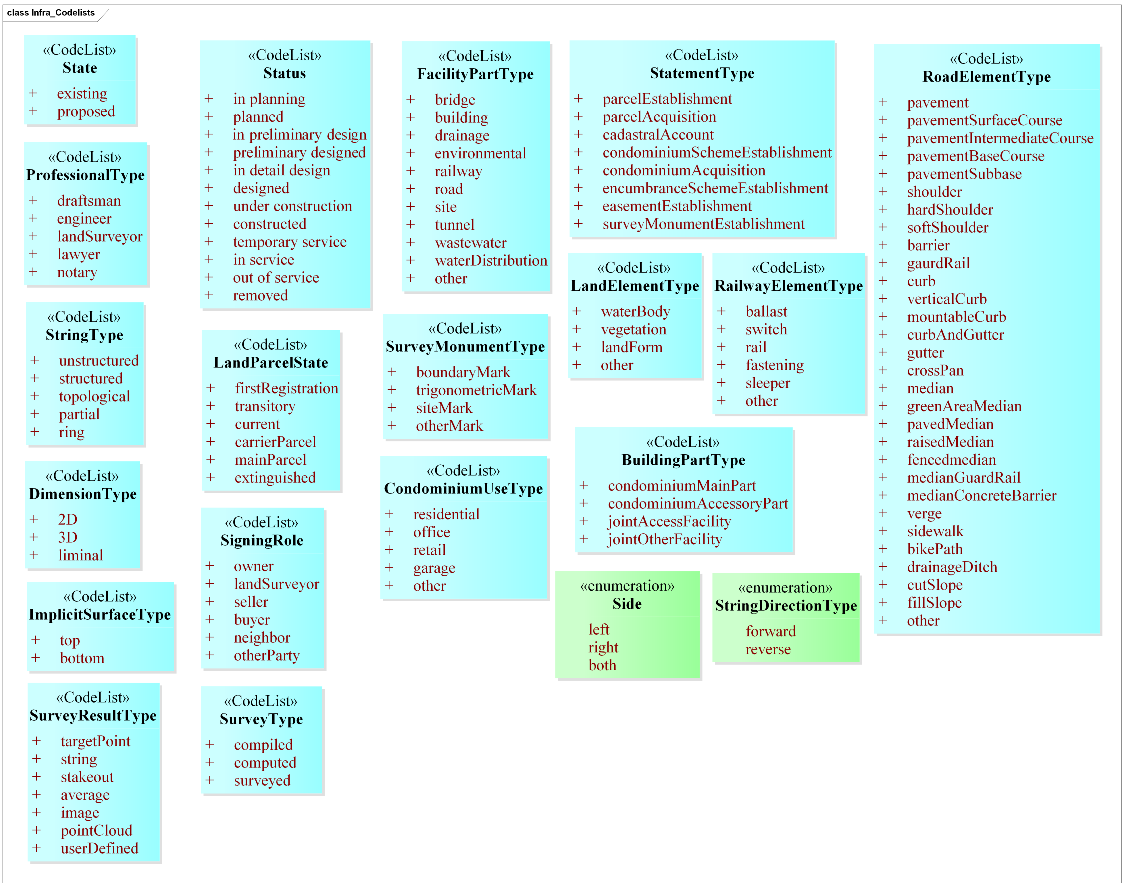


Figure A7. The UML excerpt depicts the LandInfra codelists (and enumerations) implemented in the CityGML Infra ADE.

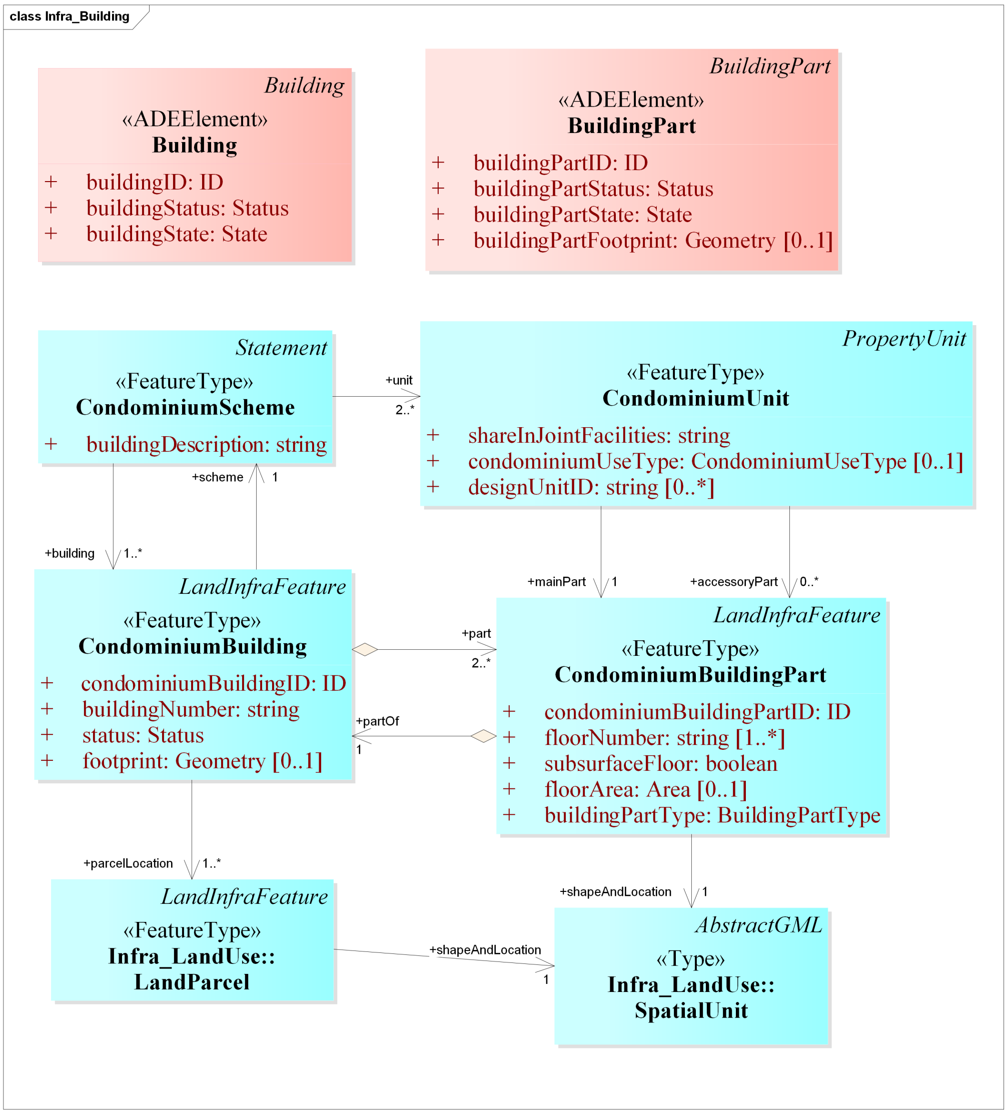


Figure A8. The UML excerpt depicts the CityGML `CG::Building` and `CG::BuildingPart` extended to include the attributes from the LandInfra `LI::FacilityPart`. It also shows LandInfra `CondominiumBuilding`, `CondominiumBuildingPart`, and other associated classes introduced in the CityGML Infra ADE.

References

1. OGC. *Land and Infrastructure Conceptual Model Standard. Doc. No. 15-111r1*; OGC: Wayland, MA, USA, 2016.
2. OGC. *City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Doc. No. 12–019*; OGC: Wayland, MA, USA, 2012.
3. OGC. *Geography Markup Language (GML) Implementation Specifications Version 3.1.1 Doc. No. 03-105r1*; OGC: Wayland, MA, USA, 2004.
4. Brink, L.; Stoter, J.; Zlatanova, S. UML-Based Approach to Developing a CityGML Application Domain Extension. *Trans. GIS* **2013**, *17*, 920–942. [CrossRef]
5. Kumar, K.; Ledoux, H.; Stoter, J. Compactly representing massive terrain models as TINs in CityGML. *Trans. GIS* **2018**, *22*, 1152–1178. [CrossRef] [PubMed]
6. Biljecki, F.; Kumar, K.; Nagel, C. CityGML Application Domain Extension (ADE): Overview of developments. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 13. [CrossRef]

7. Kumar, K.; Ledoux, H.; Commandeur, T.J.F.; Stoter, J.E. Modelling urban noise in CityGML ADE: Case of the Netherlands. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-4/W5*, 73–81. doi:10.5194/isprs-annals-IV-4-W5-73-2017. [CrossRef]
8. Agugiaro, G.; Benner, J.; Cipriano, P.; Nouvel, R. The energy application domain extension for CityGML: Enhancing interoperability for urban energy simulations. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 2. [CrossRef]
9. Labetski, A.; Kumar, K.; Ledoux, H.; Stoter, J. A metadata ADE for CityGML. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 16. [CrossRef]
10. LandXML.org. LandXML-1.2. 2016. Available online: http://www.landxml.org/About.aspx (accessed on 30 March 2019).
11. Lemmen, C.; van Oosterom, P.; Kalantari, M.; Unger, E.M.; Teo, C.H.; de Zeeuw, K. Further standardization in land administration. In Proceedings of the 2017 World Bank Conference on Land and Poverty: Responsible Land Governance—Towards an Evidence-Based Approach, Washington, DC, USA, 20–24 March 2017; pp. 20–24.
12. Cagdas, V.; Kara, A.; van Oosterom, P.; Lemmen, C.; Işıkdağ, Ü.; Kathmann, R.; Stubkjær, E. An initial design of ISO 19152: 2012 LADM based valuation and taxation data model. In Proceedings of the 11th 3D Geoinfo Conference, Athens, Greece, 20–21 October 2016; pp. 145–154.
13. van Oosterom, P.; Lemmen, C.; Thompson, R.; Janečka, K.; Zlatanova, S.; Kalantari, M. Cadastral Information Modelling. In *Best Practices 3D Cadastres: Extended Version*; van Oosterom, P., Ed.; International Federation of Surveyors (FIG): Copenhagen, Danmark, 2018; pp. 95–132.
14. Kalogianni, E.; Dimopoulou, E.; Quak, W.; van Oosterom, P. LADM and INTERLIS as a perfect match for 3D cadastre. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-4/W7*, 23–26. [CrossRef]
15. Kara, A.; Çağdaş, V.; Lemmen, C.; Işikdağ, Ü.; van Oosterom, P.; Stubkjær, E. Supporting Fiscal Aspect of Land Administration through a LADM-Based Valuation Information Model. In Proceedings of the 19th Annual World Bank Conference on Land and Poverty 2018: Land Governance in an Interconnected World, Land Governance in an Interconnected World, Washington, DC, USA, 19–23 March 2018.
16. Stubkjær, E.; Paasch, J.M.; Cagdas, V.; Oosterom, P.V.; Simmons, S.; Paulsson, J.; Lemmen, C. International Code List Management—The Case of Land Administration. In Proceedings of the 7th International FIG Workshop on the Land Administration Domain Model, Zagreb, Croatia, 11–13 April 2018; pp. 223–244.
17. Labetski, A.; van Gerwen, S.; Tamminga, G.; Ledoux, H.; Stoter, J. A proposal for an improved transportation model in CityGML. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII-4*, 89–96. [CrossRef]
18. Niestroj, M.G.; McMeekin, D.A.; Helmholz, P.; Kuhn, M. A proposal to use semantic web technologies for improved road network information exchange. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *4*, 147–154. [CrossRef]
19. Niestroj, M.G.; McMeekin, D.A.; Helmholz, P. Overview of standards towards road asset information exchange. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 443–450. [CrossRef]
20. Malmkvist, M.; Axelsson, P.; Wikström, L.; Bergman, O.; Nilsson, A.; Granberg, S.; Jensen, J.; Häggström, E.; Sigfrid, J.; Karlsson, K. *Alignment Deployment. Implementation Report. Verification IFC Alignment and InfraGML*; Technical Report; Nordic Project Team, BuildingSMART: Hertfordshire, UK, 2017.
21. UIC. RailTopoModel. 2016. Available online: http://www.railtopomodel.org (accessed on 28 March 2019).
22. Devys, E. Rail Infrastructure: RailTopoModel and LandInfra Interoperability. In Proceedigns of the 9th RailTopoModel Conference, Paris, France, 22 May 2018. Available online: http://rtm.uic.org/wp-content/uploads/2018/06/10-ISN-18.063-10-Rail-infra-RailTopoModel-LandInfra-Interoperability-220518.pptx (accessed on 28 March 2019).
23. Pouliot, J.; Larrivée, S.; Ellul, C.; Boudhaim, A. Exploring schema matching to compare geospatial standards: Application to underground utility networks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 157–164. [CrossRef]
24. OGC. *OGC Underground Infrastructure Concept Study Engineering Report. Document No. OGC 17-048*; OGC: Wayland, MA, USA, 2017.
25. Becker, T.; Nagel, C.; Kolbe, T.H. Integrated 3D modeling of multi-utility networks and their interdependencies for critical infrastructure analysis. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin, Germany, 2011; pp. 1–20.
26. OGC. *PipelineML Conceptual and Encoding Model Standard*; OGC: Wayland, MA, USA, 2016.

27. Blanchet, C.; Castaing, C.; Beaufils, M.; Emmanuel, D. GeoBIM (MINnD) Use Case on an Infrastructure Acoustic Study: Feedback on the Use of CityGML and InfraGML. 2017. Available online: https://portal.opengeospatial.org/files/?artifact_id=75554 (accessed on 28 March 2019).
28. Rajabifard, A.; Atazadeh, B.; Kalantari, M. A critical evaluation of 3D spatial information models for managing legal arrangements of multi-owned developments in Victoria, Australia. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 2098–2122. [CrossRef]
29. BuildingSMART. BuildingSMART for Infrastructure. 2015. Available online: http://www.buildingsmart-tech.org/infrastructure/projects (accessed on 30 April 2019).
30. Kumar, K.; Labetski, A.; Arroyo Ohori, K.; Ledoux, H.; Stoter, J. The LandInfra standard and its role in solving BIM-GIS quagmire. *Open Geospat. Data Softw. Stand.* **2019**, under review.
31. Safe Software. FME 2019.0. 2019. Available online: https://www.safe.com (accessed on 15 May 2019) .
32. KIT. FZK Viewer 5.1. 2019. Available online: https://www.iai.kit.edu/1302.php (accessed on 30 April 2019).
33. El-Mekawy, M.; Östman, A.; Shahzad, K. Towards Interoperating CityGML and IFC Building Models: A Unified Model Based Approach. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin, Germany, 2011; pp. 73–93.
34. Ledoux, H. On the validation of solids represented with the international standards for geographic information. *Comput.-Aided Civ. Infrastruct. Eng.* **2013**, *28*, 693–706. [CrossRef]
35. Ledoux, H. val3dity: Validation of 3D GIS primitives according to the international standards. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 1. [CrossRef]
36. Zobl, F.; Chmelina, K.; Faber, R.; Kooijman, J.; Marschallinger, R.; Stoter, J. Multidimensional aspects of GeoBIM data: New standards needed. In *Mathematical Geosciences at the Crossroads of Theory and Practice, Proceedings of the IAMG2011 Conference, Salzburg, Austria, 5–9 September 2011*; International Association for Mathematical Geosciences (IAMG): Houston, TX, USA, 2011; pp. 1–14.
37. ISO. *ISO 19152:2012 Geographic Information—Land Administration Domain Model (LADM)*; ISO: Geneva, Switzerland, 2012.
38. Rönsdorff, C.; Wilson, D.; Stoter, J. Integration of land administration domain model with CityGML for 3D Cadastre. In Proceedings of the 4th International Workshop on 3D Cadastres, Dubai, UAE, 9–11 November 2014.
39. Góźdź, K.; Pachelski, W.; Van Oosterom, P.; Coors, V. The possibilities of using CityGML for 3D representation of buildings in the cadastre. In Proceedings of the 4th International Workshop on 3D Cadastres, Dubai, UAE, 9–11 November 2014; pp. 9–11.
40. Kim, Y.; Kang, H.; Lee, J. Developing CityGML indoor ADE to manage indoor facilities. In *Innovations in 3D Geo-Information Sciences*; Springer: Berlin, Germnay, 2014; pp. 243–265.
41. Moshrefzadeh, M.; Donaubauer, A.; Kolbe, T.H. A CityGML-based Façade Information Model for Computer Aided Facility Management. In Proceedings of the Bridging Scales-Skalenübergreifende Nah-und Fernerkundungsmethoden, 35. Wissenschaftlich-Technische Jahrestagung der DGPF, Cologne, Germany, 16–18 March 2015.
42. McNeil, S.; Tischer, M.; DeBlasio, A. Asset management: What is the fuss? *Transp. Res. Rec. J. Transp. Res. Board* **2000**, *1729*, 21–25. [CrossRef]
43. Yao, Z.; Kolbe, T.H. Dynamically extending spatial databases to support CityGML application domain extensions using graph transformations. In Proceedings of the Kulturelles Erbe Erfassen und Bewahren-Von der Dokumentation zum Virtuellen Rundgang, 37. Wissenschaftlich-Technische Jahrestagung der DGPF, Würzburg, Germany, 8–10 March 2017; pp. 316–331.
44. Yao, Z.; Nagel, C.; Kunde, F.; Hudra, G.; Willkomm, P.; Donaubauer, A.; Adolphi, T.; Kolbe, T.H. 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 5. [CrossRef]
45. Nagel, C. citygml4j: The open Source Java API for CityGML. 2015. Available online: https://github.com/citygml4j/citygml4j (accessed on 30 April 2019).

46. Target, S. The Rise and Rise of JSON. 2017. Available online: https://twobithistory.org/2017/09/21/the-rise-and-rise-of-json.html (accessed on 28 March 2019).
47. Ledoux, H.; Arroyo Ohori, K.; Kumar, K.; Dukai, B.; Labetski, A.; Vitalis, S. CityJSON: A compact and easy-to-use encoding of the CityGML data model. *Open Geospat. Data Softw. Stand.* **2019**, accepted for publication.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).